Aalto University
School of Electrical Engineering
Master's Programme in Communications Engineering

Lauri Lääperi

Common Operating Picture of Critical Infrastructure:
System Design and Implementation

Master's Thesis
Espoo, May 26, 2014

Supervisor: Prof. Jukka Manner
Instructor: M.Sc. Jussi Timonen

Aalto University
School of Electrical Engineering
Master's Programme in Communications Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Lauri Lääperi

Title:
Common Operating Picture of Critical Infrastructure: System Design and Implementation

Date:	May 26, 2014	**Pages:**	vii + 65
Professorship:	Communication networks	**Code:**	S-38
Supervisor:	Prof. Jukka Manner		
Instructor:	M.Sc. Jussi Timonen		

Modern society is highly dependent on undisrupted delivery of critical infrastructure services such as power, water and telecommunications. Real-time common operating picture (COP) of critical infrastructure is required in order to command and control recovery from occurring disturbances. Various decision makers require the real-time incident information and the state of critical infrastructure in order to make the sound decisions affecting its operation. Additionally, critical infrastructure interdependencies are essential information for the decision makers to achieve situational awareness including current and future state of the infrastructure. This thesis presents requirements and architecture for a national common operating picture system as well as system implementation suggestions.

Requirements for the COP system are defined after the composition of modern critical infrastructure is studied and the operating environment is understood. The common operating picture is provided through a flexible brokered agent-based architecture which satisfies the restrictions presented by the critical infrastructure as an environment. Architecture components are designed in accordance with Joint Directors of Laboratories (JDL) data fusion model to allow integration of different critical infrastructure systems together. Generic agent component is customized for each source system to produce events for three different analysis components which produce meaningful objects, current state and future impact estimations from the data.

Implemented prototype system is used to test and evaluate the COP system architecture. Real world data from the intrusion detection system (IDS) and supervisory control and data acquisition (SCADA) system was used to test the integration of distinct systems. Performance of the implemented system was measured and determined to be sufficient to satisfy deployment scenario requirements.

Keywords: critical infrastructure, common operating picture, architecture, JDL data fusion model

Language: English

Aalto-yliopisto
Sähkötekniikan korkeakoulu
Tietoliikennetekniikan tutkinto-ohjelma

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Lauri Lääperi		
Työn nimi:	Kriittisen infrastruktuurin tilannekuva: järjestelmän suunnittelu ja toteutus		
Päiväys:	26. toukokuuta 2014	**Sivumäärä:**	vii + 65
Professuuri:	Tietoverkkotekniikka	**Koodi:**	S-38
Valvoja:	Prof. Jukka Manner		
Ohjaaja:	DI Jussi Timonen		

Moderni yhteiskunta on hyvin riippuvainen kriittisen infrastruktuurin tarjoamista palveluista, kuten sähköstä, vedestä ja tietoliikenteestä. Reaaliaikainen kriittisen infrastruktuurin yhteinen tilannekuva, jonka avulla useat eri päätöksentekijät pystyvät tekemään järkeviä kriittistä infrastruktuuria koskevia päätöksiä, on vaatimus häiriötilanteista palautumisen johtamiselle ja ohjaamiselle. Lisäksi kriittisen infrastruktuurin sisäiset riippuvuudet ovat oleellisia päätöksentekijän tilannetietoisuuden saavuttamiseksi johon tarvitaan ymmärrys sekä kriittisen infrastruktuurin nykyisestä että tulevasta tilasta. Tässä diplomityössä esitetään vaatimukset ja arkkitehtuuri järjestelmälle, joka kykenee tarjoamaan kansallista tilannekuvaa, sekä ehdotetaan menetelmiä järjestelmän toteuttamiseen.

Tilannekuvajärjestelmän vaatimukset määritellään kriittisen infrastruktuurin rakenteen perusteella sekä toimintaympäristön ymmärtämisen jälkeen. Tilannekuva tarjotaan joustavan agenttipohjaisen viestinvälittäjä arkkitehtuurin kautta, joka täyttää ympäristön asettamat vaatimukset. Arkkitehtuurin osat ovat suunniteltu JDL (Joint Directors of Laboratories) datafuusiomallin mukaan, mikä mahdollistaa erilaisten kriittisen infrastruktuuri järjestelmien integraation. Geneerinen agenttikomponentti muokataan kohdejärjestelmäkohtaisesti tuottamaan tapahtumia kolmelle erilliselle analysointi komponentille, jotka muodostavat merkityksellisiä tapahtumaolioita, ajankohtaisen tilan ja tulevaisuuden ennusteen kerätystä tiedosta.

Toteutettua järjestelmän prototyyppiä käytetään testaamaan ja arvioimaan tilannekuvajärjestelmän arkkitehtuuria. Tosimaailman dataa tunkeutujan havaitsemisjärjestelmästä (IDS) ja SCADA (Supervisory Control and Data Acquisition) järjestelmästä käytetään testaamaan erilaisten järjestelmien integraatiota. Toteutetun järjestelmän suorituskykyä mittaamalla voitiin todeta, että toteutettu arkkitehtuuri täyttää järjestelmälle asetetut vaatimukset.

Asiasanat:	kriittinen infrastruktuuri, tilannekuva, arkkitehtuuri, JDL datafuusiomalli
Kieli:	Englanti

Abbreviations and acronyms

ATM	Automated Teller Machine
COP	Common Operating Picture
CEF	Common Event Format
CEP	Complex Event Processing
CPU	Central Processing Unit
CVE	Common Vulnerabilities and Exposures
DCS	Distributed Control Systems
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
DoD	Department of Defence
ESB	Enterprise Service Bus
HMI	Human Machine Interface
ICS	Industrial Control System
IDS	Intrusion Detection System
IED	Intelligent Electronic Device
IPS	Intrusion Protection System
IP	Internet Protocol
IRC	Internet Relay Chat
ISP	Internet Service Provider
JDL	Join Directors of Laboratories
JMS	Java Message Service
MVC	Model View Controller
ORM	Object-Relational Mapping
PLC	Programmable Logic Controller
P2P	Peer-to-peer
RAM	Random Access Memory
RMI	Remote Method Invocation
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition

SOA	Service-Oriented Architecture
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus
UUID	Universally Unique Identifier
XMPP	Extensible Messaging and Presence Protocol

Contents

Abbreviations and acronyms **iv**

1 Introduction **1**
1.1 Background 1
1.2 Research problem 3
1.3 Research questions 5
1.4 Limitations 6
1.5 Results 7
1.6 Structure of the study 7

2 Critical infrastructure **8**
2.1 Overview 8
2.2 Industrial control systems 11
2.2.1 Supervisory control and data acquisition 14
2.3 Security systems 16
2.4 Vulnerabilities 19
2.5 Summary 19

3 System specification **20**
3.1 Information sources 20
3.2 Users 21
3.3 Requirements 23
3.3.1 Integration 23
3.3.2 Dependencies 25
3.3.3 Scalability 26
3.3.4 Visualization 27
3.4 Event 27
3.5 Data fusion 30
3.5.1 JDL data fusion model 31
3.6 Summary 34

4 System architecture **35**
4.1 Scope and requirements 35
4.2 Design choices 36
4.2.1 Agent-based 36
4.2.2 Broker 38
4.2.3 Registrar 39
4.2.4 Visualization 40
4.3 Analysis 40
4.3.1 Object 41
4.3.2 State 41
4.3.3 Impact 41
4.4 Components 41
4.5 Summary 44

5 System implementation **46**
5.1 Technologies 46
5.2 Broker 47
5.3 Components 48
5.3.1 Registrar 49
5.3.2 Agent 51
5.3.3 Analyzer 52
5.3.4 View 52
5.4 Evaluation 53
5.4.1 Integration 53
5.4.2 Performance 55
5.5 Summary 56

6 Conclusions **57**
6.1 Summary 57
6.2 Suggestions for further research 59

A SCADA snapshot **64**

Chapter 1

Introduction

1.1 Background

In 2007, Estonia was subjected to cyberattacks related to dispute about relocation of the Second World War memorial statue [1]. The dispute was between Estonians and Russians, as Estonians viewed the statue as a symbol of Soviet occupation and wanted to remove it from the centre of Tallinn. After the Estonian parliament adopted laws that allowed relocation of war monuments, the statue was moved to an another site. The relocation was strongly opposed by Russians, including the Russia's state leadership, and the dispute resulted in rioting and a subsequent network attacks. The first distributed denial of service (DDoS) attacks were targeted against websites of Estonian institutions and after a couple of days they expanded to include Estonian DNS servers, Internet service providers (ISPs) and media. Attacks lasted roughly for three weeks until the situation returned to normal, but the aftermath lasted for months and revealed that most likely Russian government were involved to the attack with some other participants [1]. Estonia's reaction was to start aggressively invest on national cybersecurity.

Another recent cyberattack that gained global attention was stuxnet worm. It was a complex attack that targeted Iranian nuclear program. It operated by disturbing nuclear research facility's centrifuges by giving them false data and causing machine malfunctions. Stuxnet was distributed via USB memory sticks and it contaminated any Windows operating system it came across. Nevertheless, it managed to remain undetected because the actual warhead did not activate until it reached specific Siemens industrial controllers that had correct model number, configuration details and program code [14].Very sophisticated structure of the virus and the tightly specified target indicated that it was not manufactured by a small but a large organi-

zation that had the appropriate resources for the task. Later it was revealed that USA and Israel were responsible of the stuxnet worm [26], although it already had been speculated widely. The attack was significant as it was the first globally recognized incident where specific target was subjected to a cyberattack in order to cause physical damage to the target system [10].

Aforementioned cyberattacks are good examples of modern cyberwarfare. Although both incidents can be categorized as cyberattacks, they differ clearly on execution and impact. More specifically, they show the transition from the traditional network attack to a genuine cyberattack where purpose is to deal physical damage to the target system. The 2007 incident is an example of the traditional network attack that utilized DDoS method, whereas Stuxnet represents a cyberattack where the purpose was to damage industrial equipment. The ability to damage physical infrastructure with cyberattacks makes the threat very dangerous and at the same time very usable in a situations where traditional military operation is not a solution. When correctly executed, the source of a cyberattack can be obscured enough to make it impossible to trace it. Additionally, cyberweapons are relatively reusable, with such a modular design, that only small part must be tailored for a specific target [11]. Because of these advantages, it is easy to understand why cyberattacks are actively used in modern global operations and pose a permanent threat to all nations.

Modern national defence must take cyber threats seriously. Finland, as well as many other nations, have addressed the threat by investing aggressively on cyberdefence. In the year 2013 Finnish Security and Defence Committee provided national cyberstrategy guidelines for different government authorities. This strategy aims to place Finland in the top country in the area of cyber preparedness by the year 2016 [28]. Objective is very ambitious but at least the strategy creates some form of organizational foundation that is necessary for national cyberdefence to function. As we can learn from Norwegians, it is very important that all relevant actors share the common situational picture and work together in an effort to build a national cyberdefence system [20]. Additionally, it is important to share information between other countries, to gain ideas and not to exclude possible future cooperation. For example, Rantapelkonen and Kosola strongly suggest that Nordic Countries should work closely together when facing cyberspace challenges [23].

Modern society is strongly interlinked and is unable to function at its normal level without few critical industry sectors, such as power, water and telecommunications. Many different industry and service sectors can be qualified as necessary for the modern society, but Ted Lewis[16] defines 11 sectors that are critical for the operation. These 11 critical industry sectors are listed

below and together they are referred as critical infrastructure.

1. Power / Energy
2. Water
3. Telecommunications
4. Banking & Finance
5. Transportation
6. Chemical Industry
7. Defence Industry
8. Postal & Shipping
9. Agriculture and food
10. Public health
11. Emergency Services

Security and Defence Committee's report point out that in order to command and control disturbances that affect critical infrastructure, all parties must have reliable and real-time operational picture of the nation's critical infrastructure [28, s. 4]. This common operating picture (COP) allows decision makers to gain a cyber situational awareness that is required in making the coordinated decisions and react to situations such as cyberattacks. Providing a COP of critical infrastructure, as a part of national cyberdefence, is far from trivial. As noted above, presented 11 sectors are mostly individually administered and privately owned. Because they are not used to cooperate together directly it is not easy to create a systems that could combine information from all sectors and produce the COP. Purpose of this thesis is to define requirements and attempt to design a systems that is capable of providing a COP of the nation's critical infrastructure.

1.2 Research problem

Security of the modern IT systems have been a long time under active scrutiny. Security mechanisms have been developed for years and many good solutions for securing both independently administered and distributed IT systems exists. Modern intrusion detection systems (IDS) and intrusion prevention systems (IPS) are able to detect malicious network traffic, although with varying success rate, and this information is utilized in command centers by security service providers.

On the other hand, much of our society's critical infrastructure is controlled by industrial control system (ICS), such as Supervisory Control And Data Acquisition (SCADA) and Distributed Control Systems (DCS), that have traditionally been separate from IT systems. These control systems are usually designed to keep production lines working and the security of the system has been a secondary issue. Because our society is constantly integrated more tightly together, usually via the Internet, this leads to situation where these vulnerable control systems can be accessed directly from the public networks. As Aalto university's report on Finnish control networks vulnerability [34] showed, there are at least 2915 various control devices accessible from the Internet. Many of them are poorly protected and utilize systems that have known vulnerabilities. To make matters worse, some of these systems are actively used to control parts of our critical infrastructure such as industrial automation systems or power control and remote control systems.

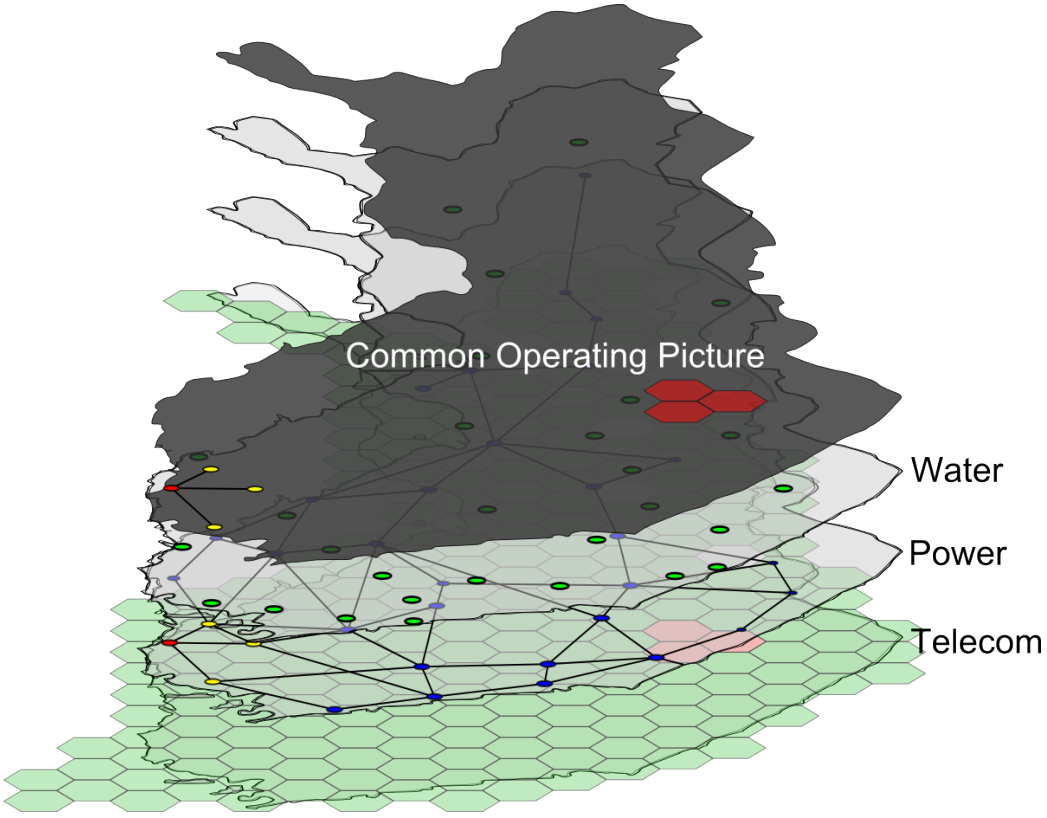

Figure 1.1: National common operating picture.

The problem from the national cyberdefence point of view is this frag-

mented infrastructure that needs to be integrated together. Construction of a common operating picture of critical infrastructure, to gain situational awareness of the system, faces integration problems because multiple different systems that are not iteroperable should work together and contribute to common knowledge, as seen on Figure 1.1. In the figure, the common operating picture is presented as composite of the different critical infrastructure actors (water and power delivery systems and telecommunication network). Common operating picture should include only relevant information to aid monitoring personnel to gain situational awareness of the critical infrastructure.

Great variety of systems and devices, that are part of critical infrastructure, pose many challenges such as interoperability, data fusion and coordination. For example, telecommunication and power delivery sectors are interleaved in such a degree that disturbances on one directly affects the other one. Therefore they must have some means to cooperate and change information. Additionally, fusing together data from systems that generate data with completely different content and volume is a challenge. For example, whereas simple control station may be working or not working, some IDS can generate vast amounts of data that may represent an intrusion to the critical system. Finally, the coordination of different authorities in times of crisis require the means to understand what is happening and how the failing of one system affects to others.

Many additional integration problems arise from the different vendors that are not allowing third parties to have access their systems. Additionally, information security laws may restrict the methods available to gather important information from the network. Regardless of these problems, a way to combine critical information from fragmented systems is needed in order to create a common operating picture of critical infrastructure.

1.3 Research questions

As mentioned in section 1.1, common operating picture of critical infrastructure is a major requirement for the national cyberdefence. A system must be designed and built to integrate together many different kinds of independent information sources to form a single national COP. Critical infrastructure has evolved to complex entity with many interdependencies and interconnections. Therefore, it is necessary to understand the environment where the COP system is operating. Limitation and requirements created by the critical infrastructure must be first acknowledged in order to understand what solutions work and what design choices are mandatory.

The first research question, this thesis should find an answers to, is what kind of system is needed to create a national COP of critical infrastructure? Defining a system specification for the system, which operates within critical infrastructure environment, is the first task in order to answer the question. The system specification should define the deployment environment as well as the requirements placed by it. Additionally, when considering the overall integration challenge we find out that the integration problem has many similarities with multi-sensor data fusion systems. Data fusion systems has been studied for many years for various applications and many well defined suitable models exist for the COP system. Therefore, the specification should also include an applicable data fusion model to be utilized within the system.

The second question for this thesis is how to implement the specified COP system? Defining specification for the system is a necessary first step on the development process but the aim of this thesis is to design a functional system that can be implemented on national scale. Therefore, an architecture which fulfills the system specification must be designed. The system architecture is a necessary step towards the system implementation. However, on its own it's not enough to answer the presented question. Therefore, the system must also be implemented in order to fully establish that the defined and designed system is able to function and provide common operating picture of critical infrastructure.

1.4 Limitations

The focus on this thesis is to define what kind of system is required to successfully integrate data from different critical infrastructure actors together. Security of the system itself is not a part of this goal and therefore most security aspects are omitted from the study. Some security related problems such as user authorization and data visibility are taken into account but the question is the system itself a vulnerable to attacks is not examined.

Another purposefully omitted topic are the legal aspects of this kind of system. As the goal is to define what the system should be able to do and how to function, we can not be restrained by the current legislature. Again, some subjects such as data privacy and confidentiality are considered, but overall they are not the limiting factors for the concept.

1.5 Results

The goal of this thesis is to define requirements and propose architecture for a system providing common operating picture of critical infrastructure. Additionally, the proposed architecture is implemented for the most part to demonstrate that (1) the architecture is feasible to implemented and (2) it can provide COP of critical infrastructure to the various decision makers. The implemented COP system was finally evaluated against the requirements, placed by the critical infrastructure as a deployment environment, and was found to be able to fulfill the requirements as well as integrate various source systems in order to provide common operating picture of critical infrastructure.

1.6 Structure of the study

This thesis continues with chapter 2 where critical infrastructure sectors and components are described. In chapter 3 we define requirements for the system, specify what are the information sources as well as user of the system and how the system can provide common operating picture for various parties. Chapter 4 presents an architecture that is required to implement the system. Different parts of the system are specified and their interactions are defined. Chapter 5 shows a one possible way to implement the system. Suitable technologies are selected and different parts of the system are implemented and the implementation is evaluated against a requirements defined on chapter 3. Finally, chapter 6 provides conclusions to the work.

Chapter 2

Critical infrastructure

This chapter provides an overview of critical infrastructure which consist of multiple sectors that are interconnected and interdependent. Complex interactions between many industry and public service actors are not trivial and therefore they are examined closely to better understand how modern developed nation's critical infrastructure operates. Additionally, a few prevalent control and protection systems within critical infrastructure are presented and studied briefly to understand how the various sectors control their operation and protect their systems. Finally, the common vulnerabilities of networked systems are highlighted.

2.1 Overview

Critical infrastructure has become the backbone of modern society as almost everything and everybody are dependent on services provided by it. Although developed nations are depending on fully operational critical infrastructure, it contains far too many vulnerabilities and weaknesses to be truly considered secure and robust. Therefore, it is necessary to identify what exactly constitutes as critical infrastructure and what the vulnerabilities are. The understanding of critical infrastructure is especially relevant from the national cyber defense point of view as one can not successfully defend something that he or she has not full knowledge of.

As presented on chapter 1, Lewis [16] separates critical infrastructure in to 11 different sectors which are power/energy, water, telecommunications, banking & finance, transportation, chemical industry, defense industry, postal & shipping, agriculture and food, public health and emergency services. Although this taxonomy describes the critical infrastructure from the United States of America point of view, it is applicable to any other

developed nation as well.

In addition to the presented taxonomy, Lewis recognizes that the critical infrastructure is highly complex with many interdependencies. Full control over this kind of interdependent system is difficult and foreseeing the consequences of rare anomalies may be impossible without more detailed information. To help perceiving the interactions within critical infrastructure Lewis presents an additional categorization of sectors in an hierarchical model. The model allocates the 11 sectors in a three level hierarchy where upper levels are dependent on the lower levels. The hierarchical model of critical infrastructure is presented on figure 2.1.

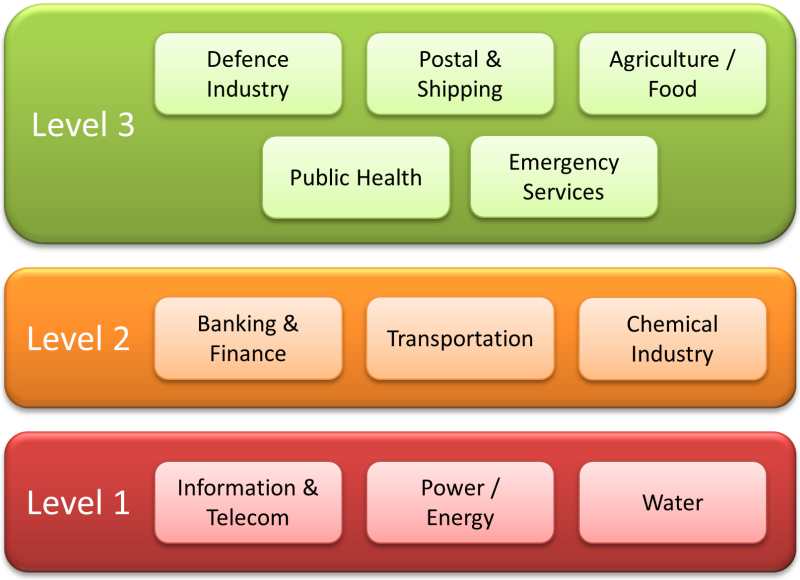

Figure 2.1: Critical infrastructure composition.

Sectors belonging to level 1 are the most critical part of the national infrastructure as they provide the means for all other levels to operate. Therefore, power, water and telecommunication sectors should be secure and resilient against threats and disturbances as all other sectors are dependent on them. Lewis' prioritized taxonomy can be utilized as a guideline to focus on the areas that are the most important for the whole critical infrastructure. It is necessary to note that dependencies within a single criticality level also exists and together with cross level dependencies are the main challenge in modeling and securing the critical infrastructure.

Lewis' taxonomy can be reasonably justified by considering peoples daily lives. Level 1 sectors are so deeply relied on in modern society that it has

become almost impossible to operate without them. Power outages can be devastating as we have no means to store electricity efficiently for long periods of time. Information and telecommunications are requirements for the normal operation of virtually every branch of society as the services are created and offered through highly networked systems. Disruptions in fresh water delivery affect large number of people especially in cities and other high population areas. In a society where bottled water in not the primary source of water the disruptions become very dangerous fast.

Justification for other levels can be similarly reasoned. For example if level 1 services would cease to exist, they would be shortly followed with the inability to pay anything, as most of the purchases are made with credit/debit cards. Even automated teller machines (ATM) are network operated and banks cash registers are not able to serve sudden surge of clients. Payment problems and power outage would relatively soon affect the transportation as gasoline pumping stations would be inoperable. Additionally, the disruptions on continuous processes of industries such as oil refineries would have a long term effect on other sectors such as transportation and energy.

Many seemingly important sectors such as public health and emergency services are placed on level 3 because of their relatively small scale impact on their absence. For example, power outages have almost instant effect on virtually every persons daily operation as most of the electronic devices cease to operate and telecommunication networks shut down soon after. Additionally, most of the payments halt as electronic payment methods stop functioning and people are unable to draw money out from ATMs. Therefore, after only an hour almost all people are affected by the level 1 and 2 service unavailability. However, the level 3 services affect a relatively small portion of people within hours of their absence.

Dependencies within a single level are also inherent in modern critical infrastructure. An example presented in figure 2.2 demonstrates that within level 1 the information & telecommunication sector is highly dependent on electricity suppliers. As power companies utilize communication networks increasingly in their processes, the dependency is established in the other direction as well. This kind of dependency network becomes highly complex when operating at the national scale. The strength of Lewis's taxonomy is that it points out the dependencies between critical infrastructure sectors and different criticality levels between groups of sectors. As all critical infrastructure actors are ultimately dependent on information & telecommunication, power / energy and water suppliers, at least some focus can be aimed at making them as secure and resilient as possible. Therefore, the taxonomy is very usable by pointing out the critical infrastructure interdependencies and the different levels of criticality.

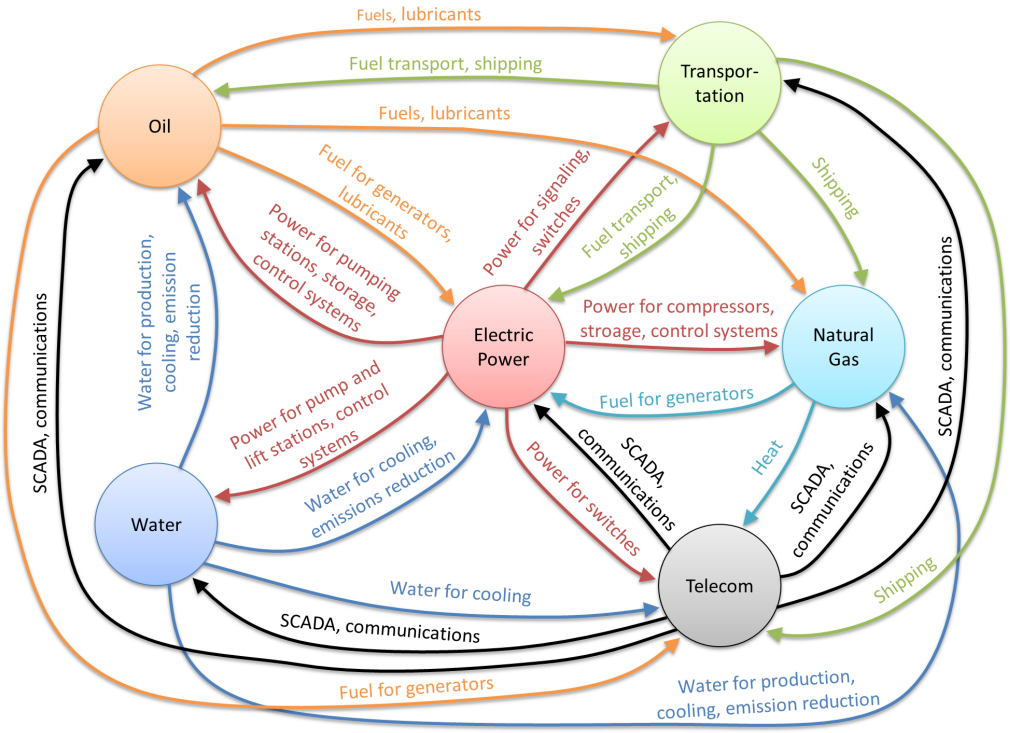

Figure 2.2: Critical infrastructure interdependencies [24].

Understanding the critical infrastructure composition at the high level is an important first step when trying to defend it from external threats. The second step from the cyber defense point of view would be to understand how the various infrastructure actors operate and provide the services which our society in so desperately dependent on. Modern networked control and monitoring devices are prevalent within all critical infrastructure sectors as they allow cost effective and flexible way to manage operation. As these systems are increasingly utilizing public networks, such as the Internet, they are allowing also network originated cyber attacks. Networked control devices play a major role in modern critical infrastructure protection and therefore they are examined more closely on next section.

2.2 Industrial control systems

Modern industrial processes have become complex and highly optimized in order to carry out their operation as efficiently as possible. Critical infras-

tructure actors, as well as any other industry actor, provide their services through these processes and are highly dependent on their continuous and correct operation. A lot of automation, monitoring and controlling equipment is needed to achieve the efficiency and precision required by modern standards. A general term for the system that is used to control various industrial processes is industrial control system (ICS).

Industrial control systems are typically used to receive sensor information from production process, automate and control the operation of a process and allow the coordination between different processes. Widely used systems include for example supervisory control and data acquisition (SCADA) and distributed control systems (DCS). In addition to previously mentioned tasks, these systems allow the management of processes remotely, between many separate operation sites. Depending on the situation the control can be centralized or distributed. Various industrial control systems have been developed to facilitate requirements presented by different production environments. Therefore, many systems that are categorized as ICSs are targeted to handle slightly different tasks and deployment environments than others. To gain some perspective, a few most widely used control systems and devices are introduced below.

Supervisory Control and Data Acquisition (SCADA) is a centralized system for monitoring and controlling various production infrastructures. SCADA focuses more on coordination of the system functions rather that controlling individual process elements. More detailed examination of SCADA is presented on section 2.2.1

Distributed Control System (DCS) is a broad term for systems where control devices are distributed instead of centralized as in SCADA. DCS allows direct control of devices as well as arrangement of devices in subsystems controlled by one control device. Task execution in distributed control systems is sequential and chronological whereas SCADA is even driven [18]. The process control in DCS is executed according to current production parameters and the past process state. The focus on DCS operation is not on alerts and unexpected process changes but running the process as intended by controlling various component to operate together.

Geographically-distributed critical infrastructure sectors such as gas, water, electrical power distribution and railroad transportation utilize widely DCSs as control mechanisms. These industries have strict requirements on parameters such as reliability and low latency as they are utilized in obtaining real-time telemetry and control real-world pro-

cesses. Therefore, adding security mechanism for the systems is difficult as they should be real-time, high-speed and have low-overhead in order to avoid interfering with operation execution. [9]

Remote Terminal Unit (RTU) is a device which gathers data from various analogue sources and converts them to digital signals. Additionally, RTUs are capable of receiving commands and forwarding them to the specific field sensors. Traditionally RTUs have been very simple devices but modern versions are capable of providing more diverse services such as higher-level processing and remote configuration [18].

Programmable Logic Controller (PLC) is a input/output device designed to control various devices in real-time environments. Usually PLCs are simple controllers containing logic and programming to allow the control of functions that do not require the supervision of process control systems. PLCs offer different number of I/O ports to connect sensors and actuators and have a specified scan rate. In addition to input and output, PLCs usually contain necessary logic to prevent device to function lower or above defined limit values and allow the continuous operation even if connection is lost to the higher level controlling system such as SCADA or DCS. Additionally, modern PLCs can have more features such as user interface, ability to receive and respond to process events, aggregate data and generate advanced reports [18].

One common feature for all industrial control systems is their purpose to control real world processes. Usually strict operation parameter requirements, such as low latencies and real-time environment, have placed the emphasis on functionality. Security aspect of these networked control systems have traditionally been only a secondary concern or ignored entirely. Additionally, these systems have utilized proprietary communication protocols which have been developed for a specific environment.

Since Internet services started emerge on the consumer markets in mid-1990s, industrial control systems have been converging with IP networks. Utilization of the Internet is constantly increasing among ICSs as the network provides an easy connectivity across great distances and flexibility on services and organization's operating structure. Macaulay [18] provides a few other reasons as why the convergence with IP networks has taken place:

- Operational costs can be reduced if physical and logical networks are shared.

- Common network technologies reduce networking costs and allow new competition between device manufacturers to produce more features and applications.
- Production is improved by new features and functions as well as allow different lifestyles and working styles.
- More product flexibility can be achieved through new services.

The insecure nature of the Internet has presented serious security concerns to these systems. Because control systems are evolved from propriety systems that have been operating in isolated networks, they contain vulnerabilities that were not previously considered. Vulnerabilities such as, lack of encryption with open network standards and hardware default passwords are problems on many legacy systems. Although latest generation of ICSs have been developed with the security aspects in mind, many critical infrastructure actors still use old control systems in daily operation.

The term industrial control system it used to describe various control systems targeted to different areas of industrial process control and management. Although these systems may have significant differences in their operating principle and architecture they all share the common goal which is to control and manage real world processes. As SCADA systems are prevalent in all industry sectors, they are a good example how engineers monitor and control the production processes consisting of co-operating production systems. Therefore, SCADA system's architecture and components are examined more closely next.

2.2.1 Supervisory control and data acquisition

One of the most prevalent industrial control system is Supervisory Control and Data Acquisition (SCADA). It is used for centralized control of various production infrastructures. SCADA is utilized in process control to support coordination of systems more than actually controlling various process elements. The responsibility of a SCADA system is to aggregate event data from different process control systems. The focus is on unscheduled events that are related to the changes in system processes. [18]

SCADA networks are deployed in many environments ranging from small home automation systems to large geographically distributed systems. A generic architecture for SCADA network, which allows the control and data acquisition for local and remote sites, is presented on figure 2.3. SCADA network can be separated to management and control network which provide

communication channel for control center and local or remote sites, respectively. Communication between control center and local or remote sites is achieved through SCADA servers and possible external mediums such as the Internet, leased lines or satellites [4].

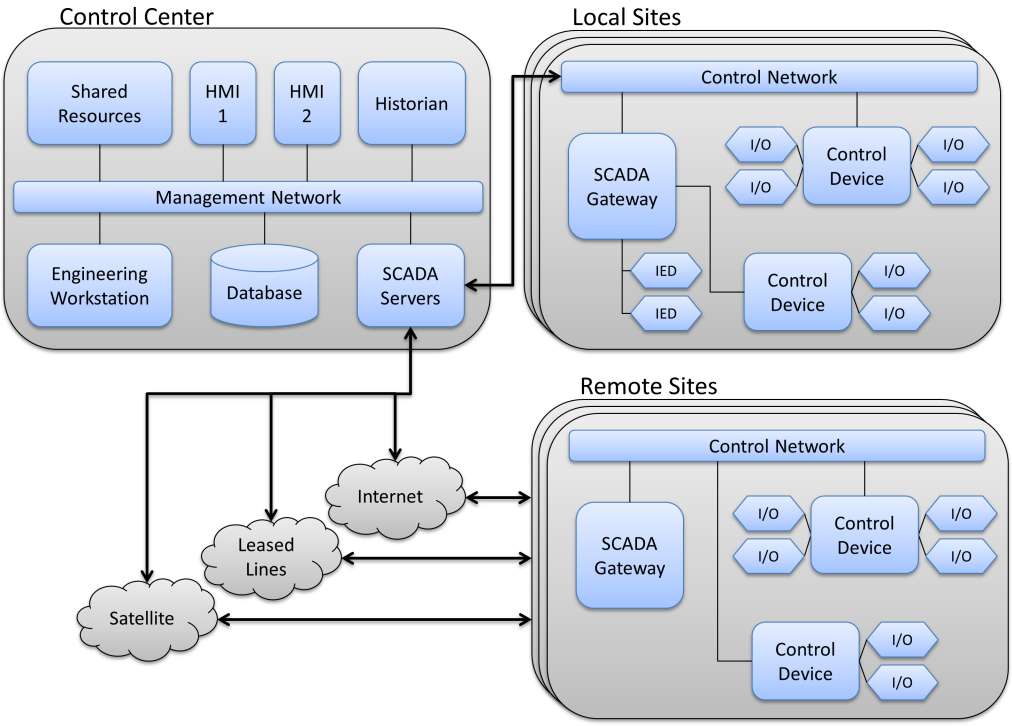

Figure 2.3: Architecture of a generic SCADA network [4].

The control center provides a connection point for all sites. The center includes devices such as SCADA servers, human machine interfaces (HMI), engineering workstations, data historians, databases and other shared resources [4]. SCADA servers manage the communication with the local or remote sites as previously mentioned. Human operators utilize HMIs to interact with the connected control and monitoring devices. Engineering station provides higher grained control over the network itself and allows the configuration of HMIs and other process control algorithms. Various process parameters and control actions are stored in the database and the history of system activities, such as network activities, sensor data and control actions, are logged by the historian. Shared resources such as printers or file servers are not directly part of the SCADA system.

Local and remote sites are connected to the control center's SCADA

servers and provide site specific control network for various devices. Control network contains components such as control devices, I/O devices and a SCADA gateway. Control devices that are able to communicate directly with the SCADA server are connected to the control network whereas other components are connected through SCADA gateway which provides an interface for the devices. Control devices such as programmable logic controllers (PLCs), remote terminal units (RTUs), input/output controllers or intelligent electronic devices (IEDs) utilize I/O devices for controlling and monitoring the process itself. Controlling and monitoring is accomplished with sensors and actuators which are capable of measuring specific process parameter and perform control actions, respectively.

First versions of SCADA systems employed a relatively primitive communication methods for linking different control components together. Data and control message transmission focused on operational requirements and didn't place much weight on security aspects because SCADA systems were physically and logically isolated from other networks [4]. As the systems evolved and adapted to new technologies through redesign and integrated components, many security vulnerabilities emerged because the old technologies were not designed to operate in the new environment.

Usually the ICSs are long time investments for companies and have operating period of decades. Therefore, the replacement of old vulnerable systems is balanced with resources and perceived threat posed by the systems. Consequently, there are currently many old and vulnerable systems in daily operation. Even within critical infrastructure sectors the amount of vulnerable control systems is alarming [34].

2.3 Security systems

Critical infrastructure sectors integrate information and telecommunication technologies increasingly with their own operations. Modern IT systems are prevalent in all industry and service sectors as they provide means to vastly improve competitiveness and productivity. Although these systems are necessary for achieving normal operational level, they are vulnerable to external and internal threats. The major incentive for companies to invest on security has been to protect their core knowledge and ensure continuous operation without disruptions. Consequently, modern security requirements are usually understood and protection systems such as intrusion detection systems (IDS) and firewalls are commonly in use to protect company's networks.

While ICSs provide information of the industry's ongoing processes, modern IT security systems can offer more detailed information of the current

level of security and possible attacks targeted to the core systems. As IT systems are currently inherent part of critical infrastructure, data provided by them should be utilized when trying to gain full situational awareness of critical infrastructure. Figure 2.4 presents simple network containing both firewall and IDS to provide security for a small organization. As the firewalls and IDSs are the basic security components in today's networks they are examined more closely next.

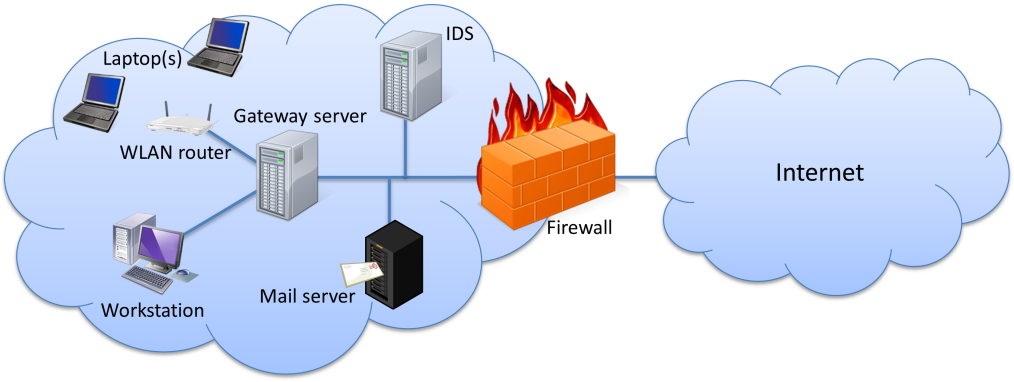

Figure 2.4: An example of a small organization's network.

Firewall

Firewall is a basic operational security component found in almost every protected network. It is usually combination of hardware and software and it operates by allowing some packets to pass through and blocking others. The purpose of a firewall is to separate organization's network from the insecure network it is connected to. Moreover, a firewall provides means to control what traffic is allowed to gain access to the administered network.

Firewalls can be classified into three categories by their operating principle: traditional packet filters, stateful filters and application gateways [13]. Traditional packet filters inspect each packet separately and determine if the packet should be allowed to enter the network. Inspection is based on packet header data such as IP addresses, port numbers, used protocol, etc. Traditional packet filters are restricted to examine each packet in isolation one at a time. Stateful filters on the other hand monitor TCP connections over a established sessions. By monitoring sessions, the filter is able to distinguish and allow packets that belong to a connection which is initiated from the secure network. Finally, the application gateways are servers that manage

all application specific traffic. They allow access to external network only from selected applications and block others.

Firewalls are an important part of all networks as they provide the most basic protection by restricting the access to the organization's network. Firewalls are usually the first devices in protected networks that can detect suspicious network anomalies and therefore can provide the first alarm of such events. For example suspicious port scans and denial of service attacks can be detected on firewall an reported onwards. Firewalls can provide valuable information for common operating picture and therefore should be utilized as information sources.

Intrusion detection systems

Intrusion detection systems (IDS) provide more thorough packet inspection services than firewalls. As firewalls are limited to inspect data from packet headers, IDS's are capable of inspecting the packet content. This deep packet inspection is required in order to detect transmitted malicious data such as a virus. One major difference to firewall is that IDS do not prevent the data traffic if anomalies are detected but only report their findings for example to network administrator. Systems which provide reporting and filtering services are called intrusion prevention systems (IPS)[13].

IDSs can be classified by the way they detect attacks as either signature-based systems or anomaly-based systems [13]. Signature-based systems inspect the content of a packet or packets and compared the data against signatures corresponding to known attacks. Signatures are generated usually by professionals who know how the attacks are implemented. The major problem with signature-based systems is the requirement for the signatures which means that the system is unable to detect yet unknown attacks. Anomaly-based systems do not suffer from this weakness. Instead of signatures they monitor normal operation of a system and try to detect anomalies in packet streams such as unusual amount of ICMP packets or port scans. Although they are able to detect unknown attacks, detecting statistically unusual traffic from normal traffic is a major challenge [13]. Therefore, most of currently used IDSs are primarily signature-based.

Intrusion detection systems are currently a hot research topic as better and more accurate detection algorithms and method are required. Currently the major problem with IDSs is the high false positive alarm probability. In high traffic environments too many false alarms make the system unusable because actual attacks can not be identified. However, IDSs are already prevalent and are able provide large quantities of usable information of many different ongoing attacks. Therefore, their output is good input for the com-

mon operating picture system.

2.4 Vulnerabilities

Presently, threats directed towards the critical infrastructure are not limited to traditional system failures, human errors or natural disasters such as storms. Because telecommunication networks are integrated to all levels of critical infrastructure new threat in form of a cyber attacks is also presented. As the cyber threats focus on affecting critical infrastructure through telecommunication networks they are operating through networked control devices. These remotely controlled devices play a major part in modern critical infrastructure protection.

Previously introduced industrial control systems, such as SCADA and DCS, have traditionally been separate from IT systems. These control systems are usually designed to keep production lines working and the security of the system has been a secondary issue. Because our society is constantly integrated more tightly together, usually via the Internet, this leads to situation where these vulnerable control systems can be accessed directly from the public networks. As Aalto university's report on Finnish control networks vulnerability [34] found, there are at least 2915 various control devices directly accessible from the Internet. Many of them are poorly protected and utilize systems that have known vulnerabilities. To make matters worse, some of these systems are actively used to control parts of our critical infrastructure such as industrial automation systems or power control and remote control systems.

2.5 Summary

This chapter presented the composition of modern critical infrastructure. Section 2.1 showed that critical infrastructure has become a highly interconnected system where almost every sector is dependent on one or more sectors. Networked control systems, presented on section 2.2, create new types of vulnerabilities to the critical infrastructure which must be understood in order to protect and have resiliency against emerging threats. Viewing of the critical infrastructure as an dynamic environment is required in order to design a system that tries to integrate information from various source systems. Therefore, knowledge of the composition, interconnections and components of critical infrastructure must be utilized in the development of common operating picture system.

Chapter 3

System specification

This chapter provides a specification for the common operating picture system. Firstly, various available information sources within critical infrastructure are recognized and the targeted users for the system are defined. Secondly, the main requirements placed by the critical infrastructure are identified and recognized. Additionally, a data fusion model which is able to successfully integrate various source systems together is proposed. As a whole, the chapter provides guidelines for the subsequent architectural design.

3.1 Information sources

Modern society is greatly dependent on the services provided by its critical infrastructure. Sectors such as telecommunications, water and power delivery are the key sectors required to sustain normal operation. These three with other eight sectors presented on figure 2.1 provide the basic services for the society. Although the sectors are vital to the functioning society, there is not currently an established means to view the state and complex interdependencies of critical infrastructure as a whole. The need for a system capable of presenting common operating picture (COP) of critical infrastructure is therefore substantial. As mentioned in chapter 1, the COP of critical infrastructure is a requirement for the functional cyber defense. Additionally, whole society and many decision makers within critical infrastructure could benefit greatly from the COP. High level decision makers could make precise decisions regarding the whole infrastructure as well as individual actors could coordinate their operations and prepare for possible problems.

The previously mentioned 11 different critical infrastructure sectors and their control and security systems are the main sources of information when constructing the common operating picture. Most of the data from the sec-

tors is produced from the systems presented on sections 2.2 and 2.3. Additionally, many other networked devices can provide valuable information and could be utilized on monitoring the critical infrastructure state. Similarly, the information gathering doesn't need to be limited on critical infrastructure actors but can span over all possible sources in order to provide more detailed overall system state.

Industrial control systems are utilized in roughly similar manner in various environments within critical infrastructure. However, the ICSs are only one type of information sources and systems such as communication networks and financial sector produce vastly differ data both in quantity and content. Therefore, it is necessary to allow heterogeneous data from source systems because too much information is lost if all data is strictly formalized in common variables. Consequently, the COP system must utilize the variable data and allow the system experts to determine what incidents are critical for their systems.

A system, trying to integrate together all critical infrastructure actors with varying priorities, needs to take seriously the question of why would the different actors participate to the common operating picture system. Participation, if not legally required, would require voluntary investments in resources and time to successfully adopt the system. Therefore, the incentive to participate should come within the organization itself. The main goal for achieving this is to produce the service in a way that participants can receive value through the utilization the common operating picture by themselves. It would be beneficial to many critical infrastructure actors to have real-time information of the surrounding services which they are dependent on. The state information can be shared to the all participating parties and thus create an incentive to participate and provide data to the common operating picture system.

3.2 Users

In order to successfully develop a system that provides value for the various end users, it is important to understand how and by whom the system would be utilized. Various users and the deployment environment guides the design parameters in all system levels from architecture to user interface. Critical infrastructure as an environment sets many requirements which must be accounted for. Additionally, the requirements set by various users must be considered. For example, critical infrastructure actors have different values and priorities regarding the security of their systems or information sharing with other actors.

The main goal for the common operating picture system is to improve situational awareness of it's users. One widely accepted definition for situational awareness is Endsley's model (Figure 3.1) which consist of three different levels: perception of elements in current situation, comprehension of current situation and projection of future status [6]. According to Endsley, these three steps are required for the person to be able to gain situation awareness in dynamic systems. Consequently, the common operating picture system should provide means for the user to be aware of various incidents occurring within critical infrastructure, understand the current state and have an projection of possible future status of observed part of the critical infrastructure.

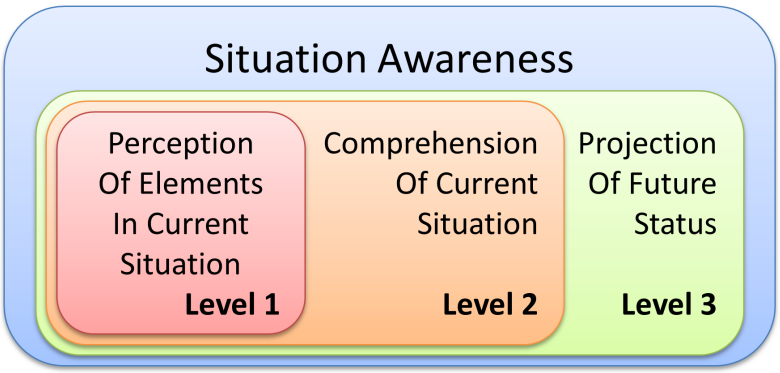

Figure 3.1: Required levels to achieve situation awareness [6].

Endsley's situational awareness model is applicable to every person utilizing the system. However, the users of the system are not a homogeneous group of people but consist of diverse selection of personnel in various organizations with different backgrounds. Additionally, the system is targeted to supports decision making on many different authority levels ranging from single critical infrastructure actor to government officials. Although the user base is broad, it can be divided to two different groups, system operators and authorities. These user groups contain the target users for the common operating picture system and are the main focus from the COP system point of view. The difference between system operator and authority user groups is defined with more detail below.

System operator is the person who is utilizing the common operating picture system to gain information corresponding on ones own system. The one feature that system operator has over the authorities, is the

ability to directly control own system. For example, in production plant the system operator could monitor their own system and other critical infrastructure actors which they are dependent on. The operator has the ability to adapt their own operations according to the occurring situation by configuring their own system or preparing for the possible future problems such as power outages or resource transportation problem.

Authorities are users that are making relatively high level decisions concerning the various critical infrastructure actors. Authorities do not have a direct control access to the source systems but monitor and react to occurring situations within critical infrastructure. On this level the common operating picture system can provide valuable information for example public safety services to react quickly and efficiently to current incidents and prepare themselves for the future situations.

3.3 Requirements

New technologies and systems are constantly introduced within critical infrastructure sectors to manage their operations. Different sectors are becoming increasingly interconnected and technologies such as modern networks are integrated to all levels of the infrastructure. Continuously evolving systems create many restrictions for the information gathering and integration. In order to accomplish the information integration the system itself should be to some extent service oriented [12, s. 57]. In order to further determine the requirements for the system the Figure 3.2 presents an outline of the COP system placement in the information loop.

3.3.1 Integration

The problem from common operating picture point of view is the fact that different critical infrastructure industries are individually administered and usually do not have means to share data between each others automatically. Additionally, the various critical infrastructure systems produce data which varies greatly in quantity and content. This kind of environment sets strict requirements for the system intended to collect and combine data from various sources. Another problem is that many automation and control systems are closed and vendor specific and as such may not easily be modified by the customers. Therefore, information from the systems need to be gathered in a way that requires and expects as little as possible from the target system.

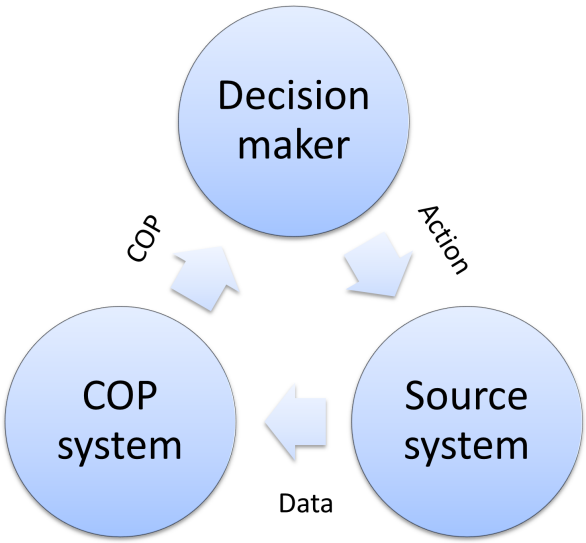

Figure 3.2: Information loop.

Additionally, it is not feasible to pack raw data from each system together as we easily loose the system specific knowledge.

Operating critical infrastructure is a system where various incidents occur constantly. Consequently, active monitoring can be achieved by collecting and analyzing critical infrastructure incidents. A specified communication format is required in order to collectively handle the incident data. Therefore, source systems should generate events from the incident data and forward them to the COP system. Events should have a common format which is independent from the observed system and they should include information such as the incident properties and how severely it affects the source system. As the systems vary greatly from each others the format should be flexible and allow various types of data to be transmitted. Additionally, the format should be easily extensible to allow functioning in evolving environment. The event format and content is defined with more detail in section 3.4.

As the focus of the common operating picture system is the information collection from the critical infrastructure actors, there is no need for the control channel to the source systems. As Figure 3.2 presents the actions of decision makers bypass the COP system entirely. Therefore, it is sufficient that information can be collected from the source systems and the control channel can be separated from the system itself. The source system control should be excluded from the COP system for many reasons such as information security, vulnerabilities, vendor specific systems and overall system

complexity. Source system control should always be handled by the local operator/administrator as they have the knowledge of how their system operates an what actions are needed in various situations.

Although the information can be collected from the critical infrastructure as events, the analysis of the diverse data sources is not a trivial problem. Even if the format is common for all events the content can vary greatly between different source systems. Therefore, analysis of the events present a data fusion problem which must be solved in order to form a coherent COP of the critical infrastructure. Fortunately the data fusion field can provide many models to help fusing heterogeneous data together. One model that is extremely suitable for situational awareness system is JDL model and it is examined more closely on section 3.5.

3.3.2 Dependencies

One major requirement for the system is to handle interdependencies existing within critical infrastructure. As the critical infrastructure offers services that are vital to the society, it is necessary to be able to present the interaction of different actors in a way that supports situational awareness. With proper tools the complex dependency network can be obtained within the critical infrastructure and it can be subjected to comprehensive scrutiny. The primary goal for the COP system is to offer tools to examine the state of critical infrastructure where the dependencies are a crucial part of the analysis.

Although source system generated events are good for monitoring real-time incidents within critical infrastructure, they do not provide a good way to deliver dependency information from the various systems. Few fundamental differences between events and dependency information exists that suggest different handling mechanism for them. Firstly, the existing dependencies are somewhat static and do not change frequently. Therefore, there is no need to send real-time information to define them. Secondly, they are usually not associated with occurring incidents but define the system's place among other systems. Consequently, dependency information should be obtained through other means than through events.

One clear restriction affecting the storage of dependency information is that access to data should be restricted at least on some level. It could be a serious defensive vulnerability to allow possible attackers free access to current and up to date information of critical infrastructure dependencies. Reason for active reconnaissance is to find out this kind of information and the available dependency database would offer just that to the malicious actor. Consequently, although information should be accessible to relevant parties

the access control and information restriction should be implemented for outside users. Additionally, even within allowed parties everyone shouldn't be able to see all dependencies but only the set that affects them directly.

3.3.3 Scalability

Scalability is a major requirement for the common operating picture system as the goal is to allow nationwide implementation. The more there are information sources in the system the more detailed and current overall situation can be constructed. Therefore, the system must be designed in a such way that adding new information sources to the system increases resource requirements as little as possible. Network utilization between source systems and analysis components should be flexible enough to allow traffic load sharing between multiple servers.

In accordance to the JDL model, source preprocessing acts as a first level filter to the system. The level 0 has two important roles from the scalability point of view. Firstly, preprocessing allows the addition of new source systems which may be considerably different from the other ones. The level 0 component is responsible for gathering the data from the system and generating events which have common format within the COP system. Therefore, the COP system can scale to support vastly different systems that may produce data in highly distinct ways. Secondly, the preprocessing acts as a low pass filter when it analyses and categorizes the source systems raw data. The rationale behind filtering is that only relevant events that affect the operation of the system itself and the others that are dependent on it are reported. This is especially important when the system is operating at a national scale and the amount of event data could easily overwhelm the system servers.

Although core analysis components of the system are affected by the number of source systems that produce events to the system they should not be the bottleneck. As the JDL model levels 1 to 3 are all able to continue the filtering of the input data they can retain the traffic volume on such levels that the core services are not congested. Especially level 1 object refinement has an important role as it is the first analysis component handling the events. Although the filtering can reduce the load to other levels, the level 1 should support load balancing to multiple servers. As the level 1 analyses more about individual source systems than dependencies between them, it is possible to separate the processing to multiple independent servers.

3.3.4 Visualization

The user base division to operators and authorities presents clear requirement for supporting different levels of data visualization. Various decision makers require different pieces of data and the common operating picture system should be able to offer them for each user. High level decision maker does not need to have detailed information of the occurring incidents as he has no mean to react to them. For example, if a power station generates an event concerning a failed transformer the authority should receive the information only if the incident greatly reduces the power plants ability to provide power to other critical infrastructure actors. However, the system operator is definitely interested in this kind of information because he can react to the incident by notifying maintenance staff.

In addition to incident filtering, the dependency information should be presented with different levels to the users. For the system operators it is only necessary to show states of those critical infrastructure actors which the operated system is directory dependent on. On the other hand, authorities require access to the dependency information across the critical infrastructure in order to construct a operating picture of whole critical infrastructure. This kind of restriction to the dependency information directly requires that user authorization mechanism are employed.

Although allowing users to view the available information from multiple points of view, it is also important to remember the diverse background of the users. With common operating picture system the user base consists of users with varying technical understanding. For example, high level decision makers may not be experts of the source systems and only rely on general information depicting the overall critical infrastructure state. On the other hand, systems operators utilizing the COP system may be very interested on more detailed information and have a good knowledge of their source system and its properties. Therefore, the goal for the visualization should be to allow the display of data on varying technical levels. How to visualize the COP in a correct way is not covered within this thesis, but remains as a requirement for the COP system to allow possibility for multiple different ways of presenting the data for the users. More detailed study of how the common operating picture should be visualized is presented on paper [25].

3.4 Event

Various incidents occur constantly within dynamic systems such as critical infrastructure. Different systems produce different types of incidents and the

severity of their effects is mostly system specific. Occurring incidents within critical infrastructure can be expressed as events, produced by critical infrastructure actors. Events should be the main communication format between common operating picture system and source systems. In order to efficiently process the events, they should contain information such as incident severity, category and other relevant data. Dependency information is not included in events but are obtained through other channels. The following definition can be used to describes the event within common operating picture system.

Definition 1. *Event is an action or occurrence detected by a program or operator that is relevant in the operation of the observed system or environment.*

As noted on section 3.1, critical infrastructure source systems produce data on vastly different quantity and content. Because the systems are different and have varying priorities the event content should be as flexible as possible. As source systems can utilize the common operating picture system by them selves the event data should not be restricted to a common content. The Ideal format for the flexible data content is key-value pair element list where reserved keywords are specifies from the COP system and additional data element can be used freely. Consequently, common data elements can be utilized by the COP system as well as other monitoring parties and additional field can be used by the source system operators them selves.

Table 3.1: Event severity

Severity	Description	Performance reduction (%)
Very High	Performance of the system is severely degraded.	80-100
High	Performance of the system is significantly degraded.	60-80
Moderate	Performance of the system is moderately degraded.	30-60
Low	Performance of the system falls below objectives but remains well-above minimum requirements.	10-30
Very Low	Performance of the system is not severely affected.	0-10

Few fixed parameters are needed for events in order to process them efficiently. Parameters that are relevant from the common operating picture point of view are event severity and category. These two parameters

are mandatory for every event. Event severity describes how severely the occurred incident affects the source system operation. Within critical infrastructure the interest is in how well the system is capable to continue it's operation and therefore the focus is on performance reduction. An event severity scale focusing on system's performance is defined on article [7] and presented on table 3.1.

Table 3.2: Event categories

Category	Description
Unauthorized Access	Unauthorized physical or electronic access to a computer system or network
Denial of Service	Any action which results in loss of system or network services normally available to authorized users. (Intentional, unintentional, electronic, digital or of natural cause)
Malicious Code	Malicious code that was installed on executed on an IT system
Improper Usage	Violations of acceptable use and security policies by authorized users
Scans, Probes, Attempted Access	Unauthorized System, network, IP, Port, Service mapping scans or probes and unsuccessful access attempts
Investigation	Events where a particular activity is suspected but unconfirmed. Once the activity is confirmed events are reassigned to the appropriate category

Another fixed parameter, event category, plays an important role in event analysis and situational awareness as it is used to understand what is occurring within monitored systems. In critical infrastructure context the categorization must contain various event types from simple hardware failures to cyber attacks. Technical document from multinational experiment 7 [21] defines a usable event categorization which is presented on table 3.2.

By using the following severity and category definition the system experts can define rules for producing events for the common operating picture system. In addition to event severity and category the key-value pairs should be reserved for the common use. The source system experts can and should

produce any information they consider important by using their own keys if necessary. Using keys outside reserved ones does not need to be reported to other parties but they can still be utilized by the operators.

3.5 Data fusion

As described on section 3.3, common operation picture system must integrate information from multiple different sources in order to help users to achieve situational awareness of critical infrastructure. The data sources in this context are information systems that are part of critical infrastructure. This includes wide variety of devices such as intrusion detection systems, firewalls, building automation systems and power management systems. Because the information sources may be very different, they produce highly distinct data that can not be integrated together directly. This problem is identical to multi-sensor data fusion systems, which Mitchell [19] defines as "*the theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representational format*". Therefore, we can utilize research on the field of data fusion to find a model that is suitable for the basis of the common operating picture system.

Data fusion systems have been studied for many years and various models and frameworks exist to guide designers. These models are applicable in cyberspace to detect intrusions and attacks against a system [3]. Choosing the right model is not trivial and it may be that no single model is suitable. As Navarro concluded [22], it is an infeasible task to try find a model that is explicit and suitable for implementation of any system. He states that one should consider data fusion models as a collection of ideas that should be combined in the implementation of a real fusion system. Consequently, the system implementation should not be the main focus when choosing the fusion model.

From theoretical point of view, there has been some research that suggest that Joint Directors of Laboratories (JDL) data fusion model is suitable and should be applied in cyber defense systems [8, 27]. Additionally, JDL data fusion model has common components with Endsley's situational awareness model (figure 3.1) and supports the situational awareness process [33]. Therefore, the JDL data fusion model is proposed as the basic guideline for the system design.

3.5.1 JDL data fusion model

The Joint Directors of Laboratories (JDL) within the US Department of Defence (DoD) has defined a framework that has become widely used in the data fusion field. Because the model is developed to aid military application development, it has qualities that make it notable in the cyber defense system point of view. JDL model describes a data fusion process that combines data from various sources in order to better understand the observed situation [31]. Fusion process consists of different levels that combine data and make inferences about the information within a context.

JDL model has been revised over the years and many papers are written on adapting the model on different applications. One article applies JDL process for cybersecurity and describes the interaction of different levels in cybersecurity domain [8]. This model review, depicted on Figure 3.3, is very beneficial to the common operating picture system. As the diagram shows, the process model consist of sensors and six different fusion levels. Sensors are various information sources, such as IDSs, firewalls or log files. The level 0 fusion process is responsible for aligning the input data in a common format that the system is able to handle. Level 1 combines the level 0 data to identify observed incidents from multiple sensors. Level 2 fusion process forms a system level perspective from the current situation, and level 3 predicts future state of the system. Level 4 process can manage the sensors and allow the fusion process refinement. Level 5 is the interface between the system and the human operator. All aforementioned 6 processing levels and sensors are needed to gain a situational awareness of the system of systems [8]. Therefore, we define and examine each part in the cyber security context more closely below.

Sensors

Cyber security sensors are devices or data sources that provide observation data from various systems. Obvious sensors would be for example network IDSs and firewalls which provide information such as logged events, intrusions, operating systems and application versions. Additionally, every source system that is part of critical infrastructure behave as a sensor from the common operating picture point of view. Consequently, the data fusion process doesn't care of the sensor is one IDS or a SCADA system spanning over large geological area.

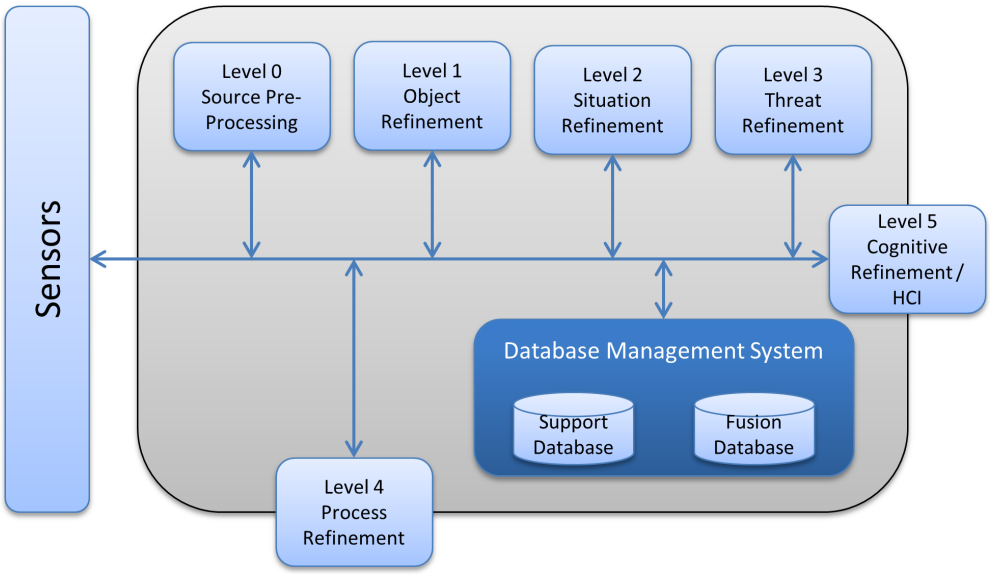

Figure 3.3: JDL data fusion process model.

Level 0 - source pre-processing

Input on level 0 process comes from the sensors and enables the system to interpret and refine the data. As mentioned above, the input for this process varies from sensor to sensor. Some of the low level data can be simple network packet captures or log files, whereas more sophisticated sensors such as IDSs can provide already refined data from the underlying processes. Regardless of the level of the input data, source data must be pre-processed to a format that can be integrated together with data from other sensors. This formatted data is the output of level 0.

Level 1 - object refinement

Level 1 processing identifies objects from a data for example by mapping IP addresses to host names. Additionally, this level collects the states of the objects. For example, commands that one user has issued could be gathered from the log file data, or alerts from multiple IDSs could be combined to create state of the attack. On the other hand, the state of the device could be just an indication if it is available or not. Output of his level should be the identified objects and their properties.

Level 2 - situation refinement

The objective of situation refinement level is to combine individual objects, created by level 1 process, and infer the current system state based on these entities. This process is the first step where situational awareness of the system is acquired. For example, the system vulnerabilities are required in order to gain a situational awareness of the system security level. Therefore, attacker's capabilities need to be recognized and combined together with current system state. Output of the level 2 fusion process must be the information about the network state, attacker's capabilities, what attacks have occurred and whether those attacks have been successful or not [8].

Level 3 - threat refinement

Threat refinement process is responsible for assessing the future impact of the attacks. Understanding the system vulnerabilities enables the process to analyze what are the attacker's options to continue the attack. Additionally, it is important that this process is aware of the available tools that are in attacker's disposal. Databases such as common vulnerabilities and exposures (CVE)[1] are useful to gain such an understanding. Output of the threat refinement should be the types of detected attacks and what are their implications.

Level 4 - process refinement

Level 4 is a meta-process that observed the fusion system takes input from outside of the system [8]. Process refinement is responsible for tasking sensors by giving them for example new IDS rules or anti-virus definitions. Additionally, the process tries to find out what sensor should be looked at more closely and what data one should try to find out. For example the operator could define that specific protocol should be given more attention and level 4 process should adapt the system to deliver more information from requested protocol. It should be noted that usually this process requires a human operator to make the decisions on where to look and what. Therefore, the process is relatively slow and situation dependent.

Level 5 - cognitive refinement

Final level of the JDL model Is the human-computer interaction (HCI). This level should provide an overview of the system state for the operator. Additionally, the operator must be able to get more detailed and filtered data

[1] http://cve.mitre.org/

from the areas of interest. Because the human operator has a crucial role in the decision making chain, in the situational awareness system, the presented views should be flexible and easy to understand. The design should take in the account the fact that operators of the system may not always be network experts but personnel that are responsible of strategic decisions.

3.6 Summary

This chapter provided system specification for the common operating picture system. Requirements and guidelines were defined for the subsequent architectural design. Four major requirements were defined: ability to integrate information from different source systems, provide the data from critical infrastructure interdependencies, scale to the nationwide implementation and finally to allow visualization of the COP from multiple perspectives to the various decision makers. Together with JDL data fusion model the specification should help to achieve a system architecture which is able provide common operating picture of critical infrastructure.

Chapter 4

System architecture

This chapter provides architectural design for a common operating picture system. The presented design complies with the requirements presented on chapter 3 and provides guidelines for system implementation. Various system components and their interactions are defined according to the JDL data fusion model as well as previously defined requirements. Proposed architecture should provide means for the implementation of common operating picture system.

4.1 Scope and requirements

The main goal for the common operating picture system is to provide comprehensive and real-time COP for various decision makers. Consequently, the system architecture must provide a platform supporting data collection, integration and analysis of different critical infrastructure sectors. Additionally, the resulted data must be presented to the users through a user interface which supports varying information levels and multiple viewpoints. This chapter presents a model architecture which is able to provide an usable platform to satisfy the requirements set for the system in chapter 3.

The main requirement for the system architecture is to provide means to integrate critical infrastructure systems together. Integration of various systems should be achieved by collecting incident data in the form of events. As the various source systems produce data that differs greatly, JDL data fusion model should be utilized in the foundation of the common operating picture system. Different JDL data fusion processes should be supported in order to successfully support situational awareness of the COP users. The source system control mechanisms are purposely left outside of common operating picture system.

Another important feature that should be supported by the system architecture is the critical infrastructure interdependencies. Interdependencies are extremely important for the common operating picture as they are the only means for understanding and determining the future state of the critical infrastructure. Therefore, the COP system must provide means for collecting dependency information from the source systems and allow the information distribution to the analysis components. As the dependency information is very sensitive, the availability must be controlled as well as security aspects ensured.

Finally, requirements such as scalability and data visualization must be supported in order to allow implementation in national scale and to support decision making for different users. The system architecture must satisfy all previously specified requirements in order to produce functional common operating picture system. Following list sums up the main requirements.

- Integrate critical infrastructure systems by collecting events and supporting all JDL data fusion model processes
- Handle critical infrastructure interdependencies
- Allow different data visualization levels
- Must be scalable in order to support nationwide implementation

4.2 Design choices

The following section present a architectural design choices that satisfy the requirements set for the common operating picture system.

4.2.1 Agent-based

Large number of different and constantly evolving source systems can not be trivially integrated together. Big data system, where raw data from the source systems is gathered and analyzed, is not feasible in this context because no one entity can understand the operation of all critical infrastructure branches. Additionally, most critical infrastructure operators are privately administered and use equipment that vendors are usually not allowing access to. The problem is reduced a little within critical infrastructure as there are only a handful of vendors that provide industrial control systems [15], but when considering the whole critical infrastructure the problem remains. Therefore, we can conclude that a system specific component is required to

collect, filter and transmit information from the source system to the COP system.

System specific component which is able to integrate various systems together can be accomplished with software agent. An agent-based architecture where each source system is integrated through customized agent software is the appropriate solution for the integration. The agent can collect information from the system and produce output that can be processed collectively. Additionally, it allows integration of different critical infrastructure systems together without losing the knowledge of the source system experts as they are utilized in customizing their agent. The responsibility of the expert is to define rules for the agent to detect important incidents that affect the system operation. Additionally, more sophisticated analysis means such as anomaly detection can be utilized in incident detection. The responsibility of agent customization should be on system experts because they have the full knowledge of the source system. Additionally, they are able to separate incidents that are critical to the operation of the system from the unimportant ones.

By default, the agent component should have low requirement for processing power and storage. However, an agent should store recent events to the local database whether or not it has transmitted them to the COP system. Reason for this, is that these event are accessible on request if needed. For example, agents that produce large amount of events, can be configured to send only events that have severity level above some threshold. If however the operator would like to examine the agent more closely it can request the previously insignificant events in order to carry out more rigorous analysis of the source system state.

The information about the state of the system is very important from the operational point of view. Maintaining the state information of each source system could be implemented on the agent, but there are few reasons why it shouldn't. Firstly, in many environments it is better to keep the agent functionality as simple as possible. If the agent focuses on simply producing events it has considerably lower complexity and requires much less administrative resources to be kept up to date. Secondly, there are situations in which the state of the agent is dependent on the external actors such as power or telecommunication providers. For example, the source system itself can be fine but is not operating because thy are affected by external interferences such as a power outage. To keep the agent complexity low it is better to keep the state of the system within the COP system instead. The main drawback in the separation of source system state and agent is that determining the system state from the event data may not reflect the reality closely enough. Additionally, if the system state is determined according the incidents there is no a direct way to determine when the system is recovered

from the failure. Despite of mentioned drawbacks the low complexity of the agent is the greater factor as the main goal for the system is to provide means for reacting occurring incident.

4.2.2 Broker

JDL data fusion model is central for the system architecture. The model's level 0 is directly comparable to the agent component and the other levels fit well in the critical infrastructure environment and support situational awareness. Although the process model itself does not take stand on architectural decisions, it defines required steps the system must be able to offer. The architecture must accommodate all six data fusion processes and allow them to work together in a flexible and scalable way. Inter-component communication channel is a key feature allowing the operation in distributed environment and implementation on national scale. Sufficient communication channel can be achieved with a common message bus.

Requirements for the common message bus is firstly to have capacity to handle large amounts of messages from multiple sources and secondly to allow the routing of message to one or multiple destinations. The message bus must allow a large number of agents to send information from their respective systems to the COP system. Additionally, in must allows flexible and scalable way for a component to communicate with any other one within the fusion chain. Role of the message bus is central to the functioning of the system. Therefore, it is important to be able to scale the capacity by distributing the load to multiple servers as well as ensure service availability by duplicating the access points.

Message bus and the components can be developed by various technologies. The bus could be implemented, for example, as an enterprise service bus (ESB), p2p network or a cloud service. The most important function of the message bus is to allow large number of agent to send their event to the analyzers. Additionally, there may be separate analysis components that require the same streams through broadcasting. It is necessary to keep the system simple to manage and especially the agent should be able to run on low end equipment. The question is, which solution is the most applicable in this kind of environment?

One of the best architectures for this kind of environment is brokered architecture. Broker can be seen as cloud service where group of servers together offer message transfer services. Various services such as broadcasting and bi-directional messaging can be offered with little overhead. Same events can be directed to multiple destination at almost simultaneously. Additionally, as most of the communication between component are fire and forget

types, it can be easily scaled to handle all communication between various system components.

4.2.3 Registrar

Dependency information of various critical infrastructure source systems is in a major role within the COP system. Handling, updating and storing of the information has many requirements that are previously described. However, the COP system needs a way of storing and sharing the dependency information for parties that are allowed to obtain and require the information.

The information storage itself can be designed to be centralized or decentralized. As different participants of the system are already committed to produce and customize their agent software the simple method would be to include the dependency information directly to agent. This would allow the privacy for the dependency information but the problem would emerge in sharing and acquiring the information. For example if only the agents are aware of dependencies how would they know which other agents they are depended on. Although distributed solutions such as P2P hash tables exist for sharing data over distributed systems, they introduce unnecessary security problems [29]. Therefore, a centralized system is the choice for registrar as it can offer better security and access control to different parts of information.

The dependency information should be stored at a centralized registrar component (Figure 4.3), where system administrator can register their agent and define dependencies to other agents. Centralized solution allows the control of information visibility to different users. Additionally, dependencies can be defined between all registered agents. Although the data is stored in a centralized registrar system the responsibility of keeping the dependency information up to date should be on the source system administrators i.e. same person who is responsible for the agent customization.

Although handling the dependencies is the primary task for the registrar, there are few other topics it is required to address. The first and most important one is agent registration and identification, which is required for separating and linking events to source systems. Because a large number of agents may be present, the id space should be large. Additionally, identifiers should be allocated randomly to make it more challenging to brute force or guess used id's. The second is the handling of user accounts that are used to operate within the system. User accounts are necessary for assigning ownership status to the agents. The registrar component should be able to provide all mentioned functions.

4.2.4 Visualization

Common operating picture system is intended for various decision makers that require varying levels of information. The exact specification for the data visualization is out of scope for this thesis but the system architecture must allow the data presentation from many different view points. Therefore, a view component is required which handles the data presentation and offer an user interface to the system.

User interface should be offered through an Web application as it is easily usable in any modern computer such as laptop, desktop computer or computer backed monitoring room. Access through Web browser makes the user interface resilient and removes the need for any additional software installations from the user. Web interface can easily provide different levels of data visibility by filtering the information allowed to various users.

View component handles the presentation of current and future critical infrastructure state to the users. The component acquires the data directly from the event streams or by requesting it from the analysis components. Because the message bus is able to offer event streams to all willing parties, the view can simply listen the generated event streams. The data requests are executed through component interfaces.

Another task for the view component is to offer users the means refine the system operations. Tasks, such as controlling the reported event severity level of the agent or setting certain incident thresholds for analysis components, should be managed through the view component. There are of course authorization limitations on who can do what, but the default principle is that owners of agents should be the only ones able to control operations that affect them.

4.3 Analysis

The core task for the COP system is to analyze agent generated event streams and create a comprehensive common operating picture from the data. The analysis component produce event on different levels, which are object, state and impact. These states follow the JDL data fusion model and handle the tasks defined in section 3.5. All analysis component are connected through the message bus and therefore can be distributed to separate servers. However the state analyzers require access to common database in order to determine state for the whole system, and the impact analyzer requires access to dependency information between different source systems. The analysis subcomponents are examined with more detail below.

4.3.1 Object

Object analyzer is responsible of handling the events that originate from agents. It analyses the event streams and filters out undesired events. Additionally, object analyzer can detect and generate new event from the received ones. For example if many different agents are reporting port scanning incidents, one larger alert which contains all targeted systems can be generated. Because object analyzer component analyzes event streams complex event processing tools should be utilized [17, 30, 35].

4.3.2 State

State analyzer forms states of all source systems based on object analysis events. Here the state is linked with agents and stored in database. State information is constantly updated and new events are generated as the state changes. Additionally, current states of the agents can be queried through the message bus by other components.

4.3.3 Impact

Impact analyzer focuses on determining the future state of the critical infrastructure. Dependency information between different source systems, i.e. agents, is required for the analysis to allow various network analysis methods to be utilized. For example vulnerability analysis can be performed to detect critical nodes or how failures propagate through the critical infrastructure. Additionally, the alarms can be quickly propagated to specific source systems to inform incidents such as power ot telecommunication outages.

4.4 Components

Previously defined architecture components such as agent, view and analyzer are the basic building blocks of the common operating picture system. Figure 4.1 depicts common operating picture system placement in the information loop which contain source systems, COP system and decision makers. The common operating picture system interacts with various source systems through an agent component which gathers data from the system. Interaction to the decision maker is achieved through the view component which presents the common operating picture to different decision makers. Control channel shown in diagram between decision maker and source system

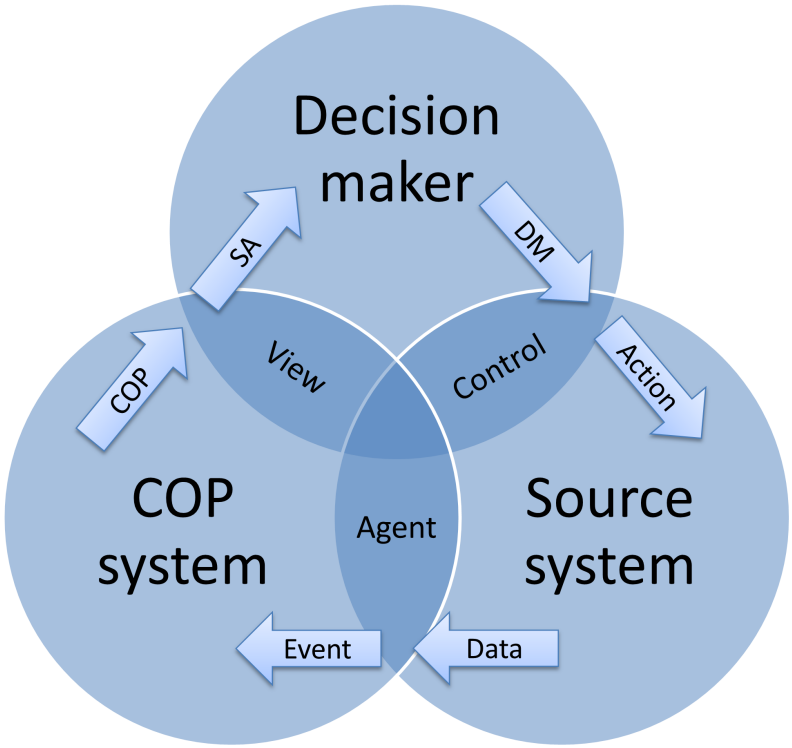

Figure 4.1: COP system placement within the information loop.

is not part of the COP system and not specified with more detail. Generally it means a direct control of system operator or possible actions from authorities.

The arrows in figure 4.1 depict the exchanged information format as the loop is traversed. Raw data from the source systems is extracted by the agent component and passed forward as events in a common format. COP system then analyses the events and creates a common operating picture which is shared to the decision makers to help support situational awareness. Decision makers then make decisions according to their perception of the situation and in a way or an another perform actions that affect the source system.

The common operating picture system's internal information flow is limited between the agent component and the view component as these are the two interfaces that connect the system to the information loop. The internal information flow can be simplified according to the figure 4.2. From the perspective of the system it is sufficient to perceive that information is received through the agent component and forwarded through the view com-

Figure 4.2: Data flow within COP system.

ponent. The one new required component is the analysis block which creates the actual common operating picture. Creation of the COP is achieved by analyzing the collected events and fusing the data with metadata such as dependency information.

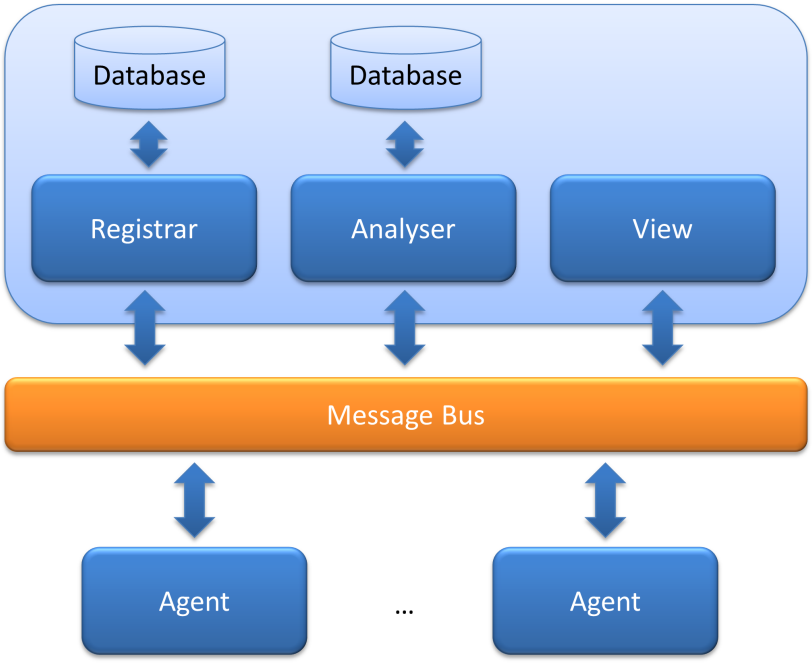

Figure 4.3: COP system components.

The system components should be as separate and independent as possible. Each component should define an interface that other components are able to use through the message bus. Interfaces allow the addition of third party services to the analysis chain which could offer more sophisticated results and complement basic functionalities provided by COP system. Figure 4.3 presents a critical system components and their relation to other components and message bus. Registrar component requires a database for the dependency information and user accounts whereas the analyzer component

requires a database for data fusion purposes.

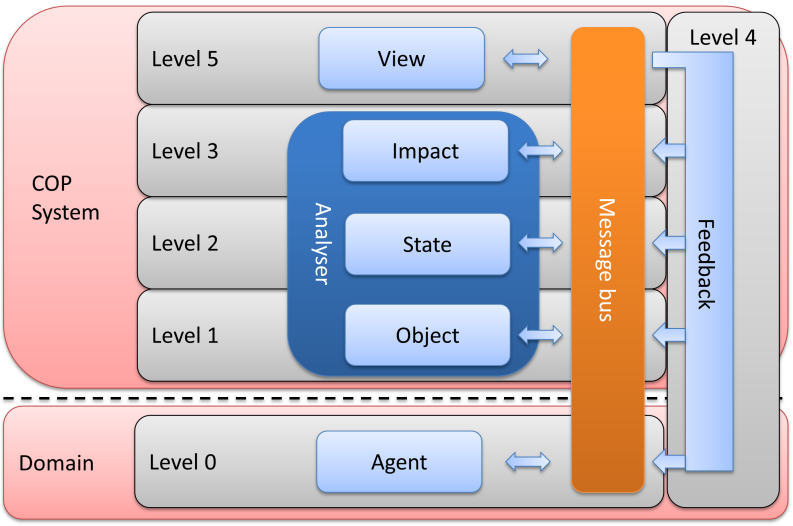

Figure 4.4: System architecture with JDL model.

Figure 4.4 depicts a logical architecture diagram for the COP system framework that follows the JDL model. All fusion sub processes are handled with respective components which communicate through the common message bus. Separation between domain and COP entities presents the administrative boundary between systems. The agent acts as a middle component between the separately administered source system and the COP system. Event analysis is separated into three different components which together, provide current and future state of the critical infrastructure. The analysis result is presented to the users through the view component in the form of common operating picture.

The message bus acts as a intermediate service for routing messages between components. It does not orchestrate the operation in any way, but only allows inter component communication. All the operation logic and actions originate from the components and users. The message bus, i.e. broker, and other presented components together form a COP system framework which allows the integration of data from separate critical infrastructure sectors.

4.5 Summary

This chapter provided the system architecture for common operating picture system. The resulted design is agent-based architecture with a brokered

message delivery service. Registrar component is utilized in agent identifier allocation as well as user account management for user authorization. Additionally, the interdependency information is handled and stored through the registrar. Analysis components are designed according to the JDL data fusion model and provide current and future impact estimations to support situational awareness. Visualization is provided through Web applications which allow flexible utilization of the system and can easily provide user specific perspectives to the data.

Chapter 5

System implementation

This chapter provides an example implementation of the common operating picture system. The system is implemented according to the architectural designs provided in chapter 4. Modern implementation technologies are chosen to allow fast development without sacrificing robustness. The system is implemented from the most part to demonstrate that the proposed architecture is feasible to implement and is able to provide common operating picture of critical infrastructure.

5.1 Technologies

The COP system was implemented from the most parts in order to test the proposed architecture. Modern programming technologies was used to allow fast development and yet to offer robust operation. Different system components are implemented using Spring framework[1] which is an open source application framework for the Java platform. Provided user interfaces utilize Spring Web MVC (Model View Controller) framework to implement Java Web applications. Web application services run on Apache Tomcat[2] server.

Implementation technology for the broker had to be chosen from multiple available candidates. The message broker which allows the inter-component communication is a central for the system architecture. Apache ActiveMQ[3] message server was chosen to implement the broker because it fully supports java message service (JMS), is scalable and released under Apache 2.0 License. Other possible broker technologies, such as XMPP and IRC, did not have as good support for transferring arbitrary objects. ActiveMQ supports

[1] http://www.springsource.org/spring-framework
[2] http://tomcat.apache.org/
[3] http://activemq.apache.org/

a large variety of client programming languages and protocols. Together with Apache Camel[4] the broker server supports Spring remoting for easy to use Remote Method Invocation (RMI). Spring remoting and RMI allow the communication between different components in as simple way as using regular java interfaces.

Many system components require a database to persist data. The implemented Java components utilize Hibernate[5] ORM (Object-relational mapping) to persist Java objects directly to the database. Hibernate supports multiple databases and therefore the selection of database is less important because the underlaying database can be easily changed to another if required. All server components utilize PostgreSQL[6] database as it is considered very stable and robust. The agent component utilizes H2[7] database as it's lightweight and can be embedded to the agent component.

It is necessary to keep in mind that although these technologies were chosen to suit the task, they are not the only possibilities for the implementation of the COP system. Many choices were affected by the previous experience of the author and on many situations the usability of the technology weighted more than performance.

5.2 Broker

ActiveMQ broker provides two different communication channels for inter-component communications, topics and queues. The difference between the two is that a topic is a type of broadcast channel whereas a queue can be compared to unicast channel. Both channels can have more than one content providers and listeners but the difference is that topic messages are forwarded to all listeners as the queue messages are forwarded to the next available listener. i.e. only one recipient. Because of these characteristics the channels are utilized in different situations.

Topics are used within COP system when a message could have more than one recipient. For example, agent generated events are transmitted to the event topic which can have multiple listeners. If an user would like to see raw unfiltered events from the agents, the view component could read the event topic simultaneously with analyzer and display the events even if the analyzer would discard them. Additionally, topics are useful when different analysis components would be implemented to listen the same even stream

[4]http://camel.apache.org/
[5]http://www.hibernate.org/
[6]http://www.postgresql.org/
[7]http://www.h2database.com/

and handle the input separately. Within COP system implementation agent generated events and analysis subcomponent (object, state, impact) events are transmitted to respective topics which allows an easy access to the data fusion chain from any component.

Queues are used for one-to-one communication. For example, the registrar service and its interface is available to other components through a registrar queue. The utilization of queue does not exclude the possibility of two different registrar servers to handle the incoming messages. However, if there would be to separate registrar servers listening the queue, the messages would be forwarded to first available registrar. This kind of system makes it easy to distribute load of slow running tasks to multiple servers. Agents utilize queues to form a control channel towards them from the COP system. Each agent is listening a queue which matches the identifier allocated to the agent. This way each operational agent will automatically be able to receive control messages such as adjustment of the threshold of reported events. All system services and agents therefore utilize queues.

On the communication protocol level the ActiveMQ supports both TCP and UDP for the message transfer protocol over the network. Within the development phase the TCP was used because the reliable connection between communicating parties was deemed more important than the performance. However, in a final system the UDP would be preferable choice to handle at least the delivery of agent generated events. UDP would not require keeping the communication channel constantly alive and would be much more robust against DoS attacks[5].

5.3 Components

System implementation follows SOA principles in a way that system composes of individual services that communicate through the broker. Various system components such as agent, registrar and analysis components are separate Java applications and do not require to be executed on a same server. Because the components can run independently and they communicate through the broker the architecture becomes very flexible. For example, in addition to agents, the registrar and analysis components can be deployed on any computer with network access and from there provide services for other component querying them through the broker. The requirement of course is the available network connection between a component and the broker.

Registrar and view components provide user interfaces through Web applications. Consequently, they are a little less independent as they require

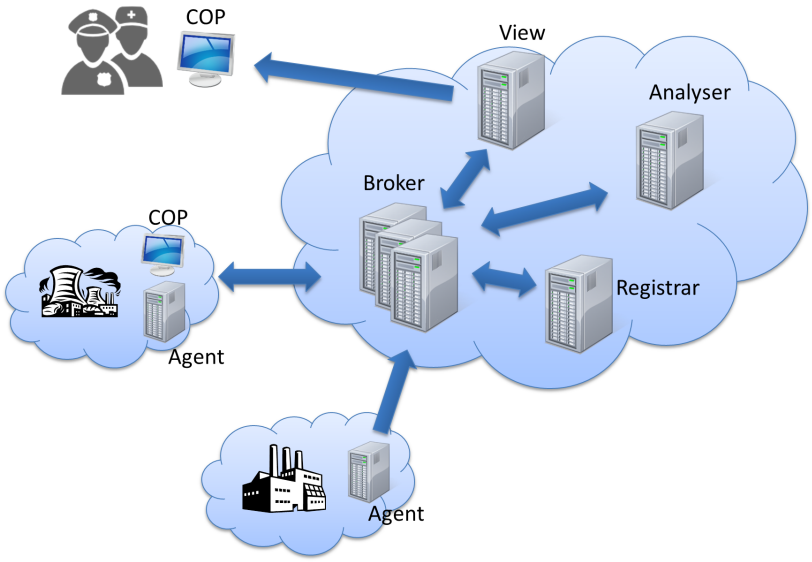

Figure 5.1: System component distribution to servers and source systems.

Java web server to be deployed on. However, they are not restricted to be deployed on the same server and can also be distributed to separate servers. Figure 5.1 presents the interactions and example distribution of different components. More detailed examination of the implementation of each system component is presented below.

5.3.1 Registrar

Registrar service is responsible of handling agents that are registering to the COP system. Registrar component is prerequisite for the system as it is responsible for the user management and agent identifier allocation. Agent IDs are realized as universally unique identifiers (UUID) which are generated when user registers an agent to the system. All agent are associated with the registered user and therefore the visibility of IDs can be restricted according to the user authorization. The registrar component offers an interface for the other component for user management, agent registration and information queries. The most relevant methods are presented below.

registerUser(User, Role)

registerAgent(Agent) : UUID

getAgents(User) : List<UUID>

getAgent(UUID) : Agent

createDependency(Dependency)

As the few above method show, users and agent can be registered through the registrar and queries can be executed for the agent IDs. Additionally, the registrar offers a method for creating dependencies between two agents. The dependency information contains the IDs of both source agent as well as the target of the dependency. Additionally, information such as the time the source agent can operate if the dependency target fails and human readable description of the dependency are stored. Registrar, as other components, is a fully independent component which is able to operate by itself.

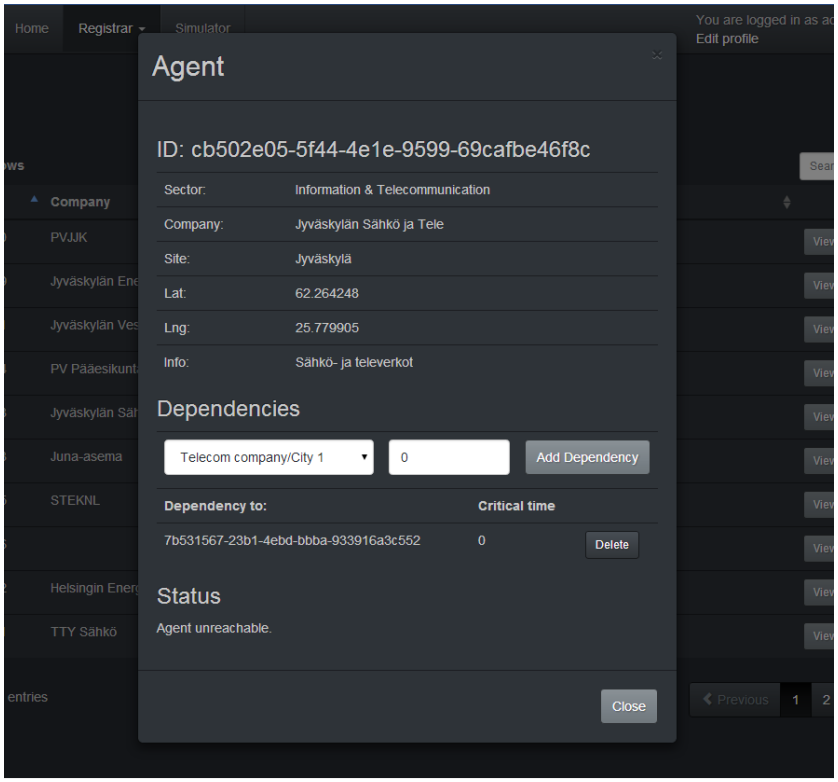

Figure 5.2: Part of the registrar's user interface presenting an information and telecommunication sector agent.

Figure 5.2 presents a screen capture from the registrar's user interface. In the figure one registered agent is selected and information such as the agent ID, sector and company are displayed. New dependencies to other agents can be defined by selecting target agent from the drop-down list and defining

critical time for the dependency, i.e. how long the system can operate if the target of the dependency fails.

5.3.2 Agent

Agent components is source system specific and therefore distributed to every site that generates data to the COP system. Produced events are sent through the ActiveMQ broker and by default received by the analyzer component. Events are encoded according to the Common Event Format [2] (CEF) in order to uniformly handle the data.

The agent is composed of two different parts, base agent and plugin. Base agent handles the tasks that are source system independent and common for all agents, such as event transmission to the COP system, timestamping and logging. The purpose of the plugin is to adapt the agent to the source system in order to allow information extraction from it. With this kind of modularization the source system expert is not required to develop the whole agent but instead can concentrate on the information extraction and determining which pieces of information are important enough to be reported to the COP system.

Utilization of the base agent is very simple. The plugin Java application is only required to create an instance of base agent and send the generated events through provided methods, as shown below.

```
//Create base agent with an ID
Agent agent = new Agent(id);

//Generate new event
Event event = new Event(cathegory, severity);

//Put extensions as key value pairs
event.addExt("lat", 60.0123);
event.addExt("lng", 24.0123);
event.addExt("descr", "Description of the event");

//Send event
agent.sendEvent(event)
```

The agent ID must be obtained manually through the registrar and provide for the base agent as a parameter. New event can be created from the source system generated data by specifying event category and severity. Additional information can be added as key value pair extensions. The event transmission is achieved as easily as calling the base agents method for

transmitting the event to the COP system.

In addition to event transmission, the base agent provides simple event logging functionality in case of COP system requires past data. The COP system can query the events generated in the past directly from the agent through the channel defined on subsection 5.2. Additionally, the base agent provides control for the threshold of event transmission. For example, if the source system generates many low severity events that are not desired from the COP system, the reporting level of the incidents can be adjusted based on event severity. In a such case the agent would only log the event and not transmit it to the broker. Consequently, the plugin does not need to be changed at all.

5.3.3 Analyzer

Analyzer component handles the incoming events which are processed according to JDL data fusion model. The component consist of three (3) different subcomponents which are object, state and impact components. Each subcomponent produces their output as events on separate broker topics which are named after the respective subcomponent. The event stream analysis itself is beyond the scope of this thesis but the presented framework provides a channels for the analysis to be carried out. Tools and methods which should be utilized in the analysis are not defined here although the Complex Event Processing (CEP) techniques seems to be promising at least in the object subcomponent [17, 30, 35].

5.3.4 View

View component provides the main user interface to the COP system. It gathers the information from other components and presents them with different perspectives to the various decision makers. The main requirements for the Web interface is to visualize current critical infrastructure state to the users in way that observer's situational awareness is improved. In order to gain full situational awareness the system should provide help on understanding the impact of various occurring incidents. As with the analysis component, the detailed specification of the user interface is beyond the scope of this thesis. However, the good specification for the Web interface is presented by Rummukainen in paper [25]. Developed setups for the user interface is presented on Figure 5.3

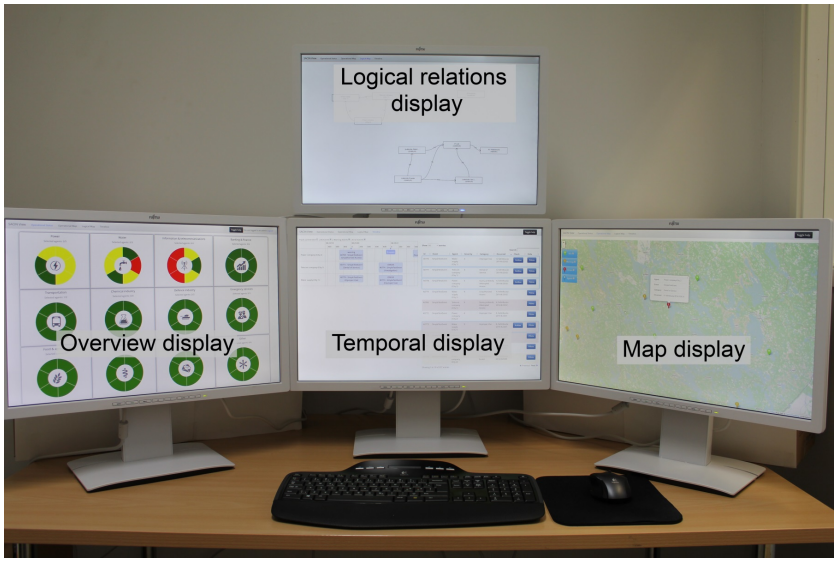

Figure 5.3: Visualization setup for the COP system [25].

5.4 Evaluation

Proposed system architecture and the implementation was evaluated against a requirements defined on section 3.3. Integration, dependencies, scalability and visualization were all taken into account in the system architecture. Table 5.1 presents how the designed and implemented system stands against the main requirements. Reflecting the resulted system against the operational scope shows that the requirements are fulfilled. For example, the integration of virtually any system can be achieved through agent based approach. Dependency information is collected by the registrar component and distributed from there to other system components. Brokered architecture allows easy load distribution and both centralized and decentralized services. Evaluation of the visualization is presented by Rummukainen [25]. Following subsection present more detailed tests for integration and performance.

5.4.1 Integration

To test the integration of various source systems few different agents were implemented. As the agents use the common base agent, it is only required to implement a source system specific plugin in order to integrate different systems to the COP system. Below is presented two different agents which utilize the same base agent and implement a system specific plugin for parsing the source system data. Implemented agent are for IDS and SCADA systems.

Table 5.1: System evaluation

Requirement	Evaluation
Integration	Various critical infrastructure systems are integrated utilizing customizable agent component. The agent generates events from the source system and transmits them to the COP system. The data fusion process follows JDL model and therefore supports all levels of situational awareness. With the agent component, virtually any system can be integrated to the COP system. The source system specific agent is able to extract information from the system according to the system administrator defined rules. As the data extraction is customizable, the source systems can vary greatly and does not need to be modified at all.
Dependencies	Interdependencies existing within critical infrastructure can be collected through the registrar. As the system administrators are the ones creating and updating the information, there is no need for an active party who is responsible to maintain data up-to-date. Obtained dependencies can be utilized in the analysis chain after they are defined by the system administrators.
Visualization	Separate view component offers the possibility to provide multiple different viewpoints from the same underlying data. Additionally, as the view component utilizes other system components and can be duplicated, there is no limit of what kind of visualizations can be created. Therefore, the architecture provides flexible data presentation support as well as means to create completely distinct visualizations if required.
Scalability	Distributed system architecture can be easily scaled if required. Broker servers can be duplicated to offer more message delivery capacity. Additionally, analysis of agent generated events on JDL level 1 (object refinement) can be separated to many servers which are able to filter events for the next fusion level. Consequently, the event filtering can be used to maintain system traffic on such levels that supports large amount of agents in distributed locations.

IDS

IDS data can be acquired directly from the generated log entries. Agent plugin in this scenario is therefore very simple as the plugin is only required to read a log file and generate new event for new reported incident. For example, Stonesoft's security management center [32] can generate log entries directly in common event format (CEF) as shown below.

```
CEF:0|Stonesoft|Alert|5.5.0|318945|MSRPC-TCP_CPS-
    Windows-MSRPC-SRVSVC-Unicode-Buffer-Overflow|10|spt
    =15743 deviceExternalId=Dubai Virtual FW 5 node 1
    dst=86.60.73.52 cat=Compromise app=Microsoft-DS rt=
    Mar 20 2014 15:09:54 deviceFacility=Inspection msg=
    Connection dropped act=Terminate proto=6 dpt=445 src
    =181.66.48.91 dvc=127.0.6.165 dvchost=127.0.6.165
    cs1Label=RuleId cs1=261002.0 cs3Label=
    VulnerabilityReferences cs3=CVE-2008-4250,BID-31874,
    MS08-067
```

The resulted event is easy to parse according to the CEF specification [2]. Additional information, such as event category, can be added by the system expert.

SCADA

SCADA plugin parses the system snapshot file generated on regular intervals from the SCADA system. The difference to the log file is that whereas the log file listener can be notified from the new lines in the file, the snapshot contains the current system state and changed element must be detected by the plugin. Therefore, the plugin must keep the state information by itself and generate new events only when new elements appear on the output for the first time. An example SCADA snapshot file with ongoing power outages can be seen on appendix A.

5.4.2 Performance

Broker performance was tested by generating events from agent and examining the event throughput. Agent was run on the same virtual server as the broker in order to minimize network latency effects on the measurements. The server resources included 2 GB of RAM and 8 CPUs with relatively small usability priority. Generated events contained information such as the event severity, category, human readable description and latitude and longitude coordinates for the occurred incident. With the included packet overhead

Table 5.2: Broker throughput

Run	1	2	3	4	5	6
Events / s	237	255	270	286	265	298

(transmission protocols and Java/Spring metadata) the event packet sizes were about 1300 Kbit. Table 5.4.2 presents the measured event throughput from different runs.

The average throughput was 257 events per second which is sufficient for the COP system deployment scenario. Consequently, considering the test server's modest resources, the measured event throughput is excellent and demonstrates that the used technologies are suitable for the system.

5.5 Summary

This chapter provided an example implementation of the proposed COP system architecture. Chosen technologies were sufficient for the prototype and should to be fitting for the implementation of the real system as well. Implemented system demonstrated the ease of integrating different source systems together with proposed agent-based architecture. Additionally, the flexible intercomponent communication and load distribution allow the implementation on national scale. Consequently, the proposed architecture is able to provide common operating picture of critical infrastructure.

Chapter 6

Conclusions

6.1 Summary

Critical infrastructure has become highly complex system of systems. Various interdependent critical infrastructure systems are separately administered but require tight co-operation in order to provide their services. Consequently, the common operating picture is required in order to command and control disturbances occurring within critical infrastructure. The main goal of this thesis was to design a system capable to provide common operating picture of critical infrastructure to various decision makers. Additionally, a system prototype was needed to demonstrate that the proposed design can be feasibly implemented.

As a deployment environment, the critical infrastructure places many requirements for the COP system. The most important ones being ability to integrate data from different critical infrastructure sectors, obtain dependency information between different systems, provide visualizations for various decision makers and offer scalable architecture to allow implementation on national scale. Additionally, identifying users and defining characteristics of various information sources are important for architectural decisions.

Useful common operating picture should improve the situational awareness of its users. Authorities would benefit greatly from the COP but so would various critical infrastructure actors. Both authorities and individual actors should adjust their operations according to their surroundings. For example, the faster disturbances within power delivery systems are acknowledged by the actors, the faster the dependent systems can react and prepare for the situation. Therefore, the incentive for the various actors to participate to the COP system should come from the utilization of common operating picture by themselves. Consequently, the system should provide tools and

visualization that take both of these user groups into account.

Understanding the information sources is another important part of the system design. The main information sources are industrial control systems within power and water sectors as well as information security systems within information and telecommunication sector. However, these are not the only sectors that are part of critical infrastructure. For example, public health and financing sectors produce data that differs greatly from the ICSs. Therefore, an event based communication with flexible key-value pair content should be used to transfer information from various systems. Additionally, the JDL data fusion model should be utilized as it offers a fusion model which complements all levels of situational awareness.

Proposed system architecture has been developed to fulfill mentioned strict requirements and to account critical infrastructure as an operating environment. Brokered agent-based architecture is scalable and offers an efficient way to integrate various source systems together. Utilization of JDL data fusion model defines necessary step to integrate various systems and provide common operating picture which improves situational awareness of users. The registrar component offers a way to collect dependency information from source systems and allowing system administrators to keep the information up-to-date.

System components should be as independent as possible. Managing individual services becomes increasingly efficient compared monolithic systems when the operating environment evolves as fast as critical infrastructure. Additionally, the brokered architecture allows multiple implementations of individual components and therefore creates possibility to develop services side-by-side providing better and more accurate results. The pros of brokered architecture can be seen also at the analysis component as the first level object analysis can easily be implemented on parallel processes improving scalability and efficiency of the whole system.

Separation of agent registration and dependency information collection to separate registrar component is an important part of the system. With this kind of architecture, the COP system administrators does not have to consist of experts from each connected source system. By distributing the administrative responsibilities to the source system experts, in the form of agent customization, registration and managing dependencies, the data privacy and administrative boundaries does not have to be compromised. Consequently, the COP system itself can focus on producing common operating picture of the whole critical infrastructure in a best way possible.

A prototype system was implemented according to the proposed architecture to confirm that the system can be feasibly implemented. Moreover, the implemented prototype system demonstrated that the architecture is ap-

plicable within the defined operational scope. Two different source systems were integrated through customized agent components and the critical system parts were implemented to create a framework supporting the creation of common operating picture. Although some parts of the architecture were not fully implemented, there should be no reason they would become an obstacle for the full system implementation.

The resulted system specification, architecture and implementation defined a system which is able to provide common operating picture of critical infrastructure. Consequently, this thesis provided answers to the research questions concerning what kind of system is needed to create a national COP of critical infrastructure and how to implement the specified system. The resulted system was found to be able to fulfill the requirements placed by the critical infrastructure environment and provide common operating picture of critical infrastructure.

6.2 Suggestions for further research

The architecture and implementation guidelines presented on this thesis allow the integration of critical infrastructure systems. The framework provides a platform for generating features such as visualization of the common operating picture and analysis of the dependencies between different systems. Therefore, further research should be directed to analyze agent generated events according to the JDL data fusion subprocesses and improve the visualization to offer better situational awareness for the decision makers.

Bibliography

[1] Alenius, K. Victory in exceptional war: The estonian main narrative of the cyber attacks in 2007. In *The Fog of Cyber Defence*, J. Rantapelkonen and M. Salminen, Eds., no. 10 in Series 2. National Defence University, Department of Leadership and Military Pedagogy, 2013, pp. 85–96.

[2] ArcSight, Inc. Common event format. [Online: `http://mita-tac.wikispaces.com/file/view/CEF+White+Paper+071709.pdf`], July 2010.

[3] Bass, T. Intrusion detection systems and multisensor data fusion. *Communications of the ACM 43*, no. 4 (2000), pp. 99–105.

[4] Chandia, R., Gonzalez, J., Kilpatrick, T., Papa, M., and Shenoi, S. Security strategies for scada networks. In *Critical Infrastructure Protection*. Springer, 2007, pp. 117–131.

[5] Chatzigiannakis, V., Androulidakis, G., and Maglaris, B. A distributed intrusion detection prototype using security agents. *HP OpenView University Association* (2004).

[6] Endsley, M. R. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society 37*, no. 1 (1995), pp. 32–64.

[7] Garvey, P. R., Moynihan, R. A., and Servi, L. A macro method for measuring economic-benefit returns on cybersecurity investments: The table top approach. *Systems Engineering 16*, no. 3 (2013), pp. 313–328.

[8] Giacobe, N. A. Application of the jdl data fusion process model for cyber security. In *SPIE Defense, Security, and Sensing* (2010), International Society for Optics and Photonics, pp. 77100R–77100R.

[9] Hieb, J., Graham, J., and Patel, S. Security enhancements for distributed control systems. In *Critical Infrastructure Protection*. Springer, 2007, pp. 133–146.

[10] Kärkkäinen, A. The origins and the future of cyber security in the finnish defence forces. In *The Fog of Cyber Defence*, J. Rantapelkonen and M. Salminen, Eds., no. 10 in Series 2. National Defence University, Department of Leadership and Military Pedagogy, 2013, pp. 99–115.

[11] Kiravuo, T. Offensive cyber-capabilities against critical infrastructure. In *Cyber Warfare*, J. Vankka, Ed., no. 34 in Series 1. National Defence University, Department of Military Technology, 2013, pp. 77–96.

[12] Kosola, J. *Development of Technology and its effects to warfare 2015-2025*. National Defence University, Department of Military Technology, 2014.

[13] Kurose, J. F., and Ross, K. W. *Computer Networking: A Top-Down Approach*, 4 ed. Pearson Education International, 2008. pp. 763–773.

[14] Langner, R. Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE Vol. 9*, no. 3 (2011), pp. 49–51.

[15] Langner, R. To Kill a Centrifuge, A Technical Analysis of What Stuxnet's Creators Tried to Achieve. Online: `http://www.langner.com/en/wp-content/uploads/2013/11/To-kill-a-centrifuge.pdf`, Nov. 2013.

[16] Lewis, T. G. *Critical infrastructure protection in homeland security: defending a networked nation*. John Wiley & Sons, 2006.

[17] Luckham, D. C., and Frasca, B. Complex event processing in distributed systems. *Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford 28* (1998).

[18] Macaulay, T., and Singer, B. *Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS*. Taylor & Francis Group, 2012.

[19] Mitchell, H. B. *Multi-Sensor Data Fusion: An Introduction*. Springer, 2007.

[20] Mørkestøl, K. H. Norwegian cyber security: How to build a resilient cyber society in a small nation. In *The Fog of Cyber Defence*, J. Rantapelkonen and M. Salminen, Eds., no. 10 in Series 2. National Defence University, Department of Leadership and Military Pedagogy, 2013, pp. 117–126.

[21] Multinational Alliance for Collaborative Cyber Situational Awareness. Collaborative Cyber Situational Awareness (CCSA) Information Sharing Framework (ISF). Tech. rep., MNE7, 2013.

[22] Navarro, A. Multisensor data fusion applied to augmented reality. Master's thesis, Delft University of Technology, 2008.

[23] Rantapelkonen, J., and Kantola, H. Insights into cyberspace, cyber security, and cyberwar in the nordic countries. In *The Fog of Cyber Defence*, J. Rantapelkonen and M. Salminen, Eds., no. 10 in Series 2. National Defence University, Department of Leadership and Military Pedagogy, 2013, pp. 27–39.

[24] Rinaldi, S. M., Peerenboom, J. P., and Kelly, T. K. Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems, IEEE 21*, no. 6 (2001), pp. 11–25.

[25] Rummukainen, L., Oksama, L., Timonen, J., and Vankka, J. Visualizing common operating picture of critical infrastructure. In *SPIE DSS 2014, to appear* (2014), SPIE.

[26] Sagner, D. E. Obama order sped up wave of cyberattacks against iran. *The New York Times* (June 1st 2012).

[27] Schreiber-Ehle, S., and Koch, W. The jdl model of data fusion applied to cyber-defence - a review paper. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2012 Workshop on* (2012), IEEE, pp. 116–119.

[28] Secretariat of the Security Committee. Finland's cyber security strategy. [Online: `http://www.defmin.fi/files/2378/Finland_s_Cyber_Security_Strategy.pdf`], Jan. 2013.

[29] Sit, E., and Morris, R. Security considerations for peer-to-peer distributed hash tables. In *Peer-to-Peer Systems*. Springer, 2002, pp. 261–269.

[30] Srinivasagopalan, S., Mukhopadhyay, S., and Bharadwaj, R. A complex-event-processing framework for smart-grid management. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2012 IEEE International Multi-Disciplinary Conference on* (2012), IEEE, pp. 272–278.

[31] Steinberg, A. N., Bowman, C. L., and White, F. E. Revisions to the jdl data fusion model. *Proc. SPIE 3719* (1999), pp. 430–441.

[32] Stonesoft Oyj. Security management center. [Online: `http://www.stonesoft.com/opencms/export/system/galleries/download/datasheets/smc.pdf`], Oct. 2013.

[33] Tadda, G. P., and Salerno, J. S. Overview of cyber situation awareness. In *Cyber Situational Awareness*. Springer, 2010, pp. 15–35.

[34] Tiilikainen, S., and Manner, J. Finnish automation network vulnerability. Tech. rep., Aalto University School of Electrical Engineering, 2013.

[35] Vranes, S., Stanojevic, M., Janev, V., Mijovic, V., Tomasevic, N., Kraus, L., and Ilic, Z. Application of complex event processing paradigm in situation awareness and management. In *Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on* (2011), IEEE, pp. 289–293.

Appendix A

SCADA snapshot

```
<?xml version="1.0"?>
<outages>
  <outage ID="106" type="fault" start="26.09.2013 10:35
     " end="26.09.2013 16:58" desc="Equipment failure."
     >
    <station code="77889" lat="62.5537927000000" lng="
       24.0668504000000" area="North" customers="9"/>
    <station code="34321" lat="62.2508442000000" lng="
       23.7804274999999" area="North" customers="5"/>
    <station code="61616" lat="61.8640298000000" lng="
       25.1910932000001" area="East" customers="5"/>
    <station code="10101" lat="61.4950159000000" lng="
       23.7773393000000" area="South" customers="10"/>
  </outage>
  <outage ID="107" type="fault" start="26.09.2013 10:05
     " end="26.09.2013 16:33" desc="Vika-alueen
     rajaaminen.">
    <station code="5604" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="18"/>
    <station code="5605" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="10"/>
    <station code="5606" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="South" customers="5"/>
    <station code="99998" lat="xx.xxxxxxxxxxxx" lng="
       xx.xxxxxxxxxxxx" area="East" customers="5"/>
    <station code="12345" lat="xx.xxxxxxxxxxxx" lng="
       xx.xxxxxxxxxxxx" area="South" customers="10"/>
```

```
  </outage>
  <outage ID="108" type="fault" start="26.09.2013 10:30
     " end="26.09.2013 18:58" desc="Sahkoverkon
     rakentaminen.">
    <station code="6160" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="12"/>
    <station code="8131" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="4"/>
    <station code="8048" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="7"/>
    <station code="8123" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="6"/>
    <station code="8049" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="12"/>
    <station code="6114" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="11"/>
    <station code="8141" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="8"/>
    <station code="8003" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="9"/>
    <station code="8112" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="North" customers="12"/>
    <station code="8058" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="North" customers="8"/>
    <station code="8053" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="North" customers="7"/>
    <station code="8047" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="West" customers="8"/>
    <station code="8016" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="North" customers="10"/>
    <station code="8059" lat="xx.xxxxxxxxxxxx" lng="xx
       .xxxxxxxxxxxx" area="North" customers="8"/>
  </outage>
</outages>
```